University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

mgr Marek Grzegorowski

Selected aspects of interactive feature extraction

PhD dissertation

Supervisor

Prof. dr hab. Dominik Ślęzak

Institute of Informatics
University of Warsaw

Auxiliary Supervisor

dr Andrzej Janusz

Institute of Informatics
University of Warsaw

May 2021

Author's declaration:
aware of legal responsibility I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

May 31, 2021 .....
date *mgr Marek Grzegorowski*

Supervisors' declaration:
the dissertation is ready to be reviewed

May 31, 2021 .....
date *Prof. dr hab. Dominik Ślęzak*

May 31, 2021 .....
date *dr Andrzej Janusz*

Abstract

In the dissertation, the problem of interactive feature extraction, i.e., supported by interaction with users, is discussed, and several innovative approaches to automating feature creation and selection are proposed. The current state of knowledge on feature extraction processes in commercial applications is shown. The problems associated with processing big data sets as well as approaches to process high-dimensional time series are discussed. The introduced feature extraction methods were subjected to experimental verification on real-life problems and data. Besides the experimentation, the practical case studies and applications of developed techniques in selected scientific projects are shown.

Feature extraction addresses the problem of finding the most compact and informative data representation resulting in improved efficiency of data storage and processing, facilitating the subsequent learning and generalization steps. Feature extraction not only simplifies the data representation but also enables the acquisition of features that can be further easily utilized by both analysts and learning algorithms. In its most common flow, the process starts from an initial set of measured data and builds derived features intended to be informative and non-redundant. Logically, there are two phases of this process: the first is the construction of the new attributes based on original data (sometimes referred to as feature engineering), the second is a selection of the most important among the attributes (sometimes referred to as feature selection). There are many approaches to feature creation and selection that are well-described in the literature. Still, it is hard to find methods facilitating interaction with users, which would take into consideration users' knowledge about the domain, their experience, and preferences.

In the study on the interactiveness of the feature extraction, the problems of deriving useful and understandable attributes from raw sensor readings and reducing the amount of those attributes to achieve possibly simplest, yet accurate, models are addressed. The novel methods proposed in the dissertation go beyond the current standards by enabling a more efficient way to express the domain knowledge associated with the most important subsets of attributes. The proposed algorithms for the construction and selection of features can use various forms of information granulation, problem decomposition, and parallelization. They can also tackle large spaces of derivable features and ensure a satisfactory (according to a given criterion) level of information about the target variable (decision), even after removing a substantial number of features.

The proposed approaches have been developed based on the experience gained in the course of several research projects in the fields of data analysis and processing multi-sensor data streams. The methods have been validated in terms of the quality of the extracted features, as well as throughput, scalability, and robustness of their operation. The discussed methodology has been verified in open data mining competitions to confirm its usefulness.

Keywords: Feature Extraction, Feature Selection, Rough Set Theory

ACM Computing Classification (rev.2012): Computing methodologies ↦ Machine learning ↦ Machine learning algorithms ↦ Feature selection.

Streszczenie

W rozprawie poruszono problem interaktywnej ekstrakcji cech (ang. interactive feature extraction) oraz zaproponowano szereg innowacyjnych podejść do automatyzacji procesu ich tworzenia i selekcji rozważając możliwość angażowania w ten proces użytkowników. Przedstawiono aktualny stan wiedzy w dziedzinie ekstrakcji atrybutów oraz zaprezentowano znane z literatury zastosowania komercyjne tego procesu. Omówiono wyzwania związane z przetwarzaniem dużych zbiorów danych, ze szczególnym naciskiem na przetwarzanie wielowymiarowych szeregów czasowych. Poddano dyskusji problem opracowania takiej reprezentacji danych, która byłaby zrozumiała dla ekspertów dziedzinowych. W tym celu, przedyskutowano możliwość wykorzystania atrybutów uzyskiwanych metodą przesuwnego okna czasowego oraz granulacji atrybutów. Opracowane metody i algorytmy ekstrakcji cech poddano weryfikacji eksperymentalnej oraz przedstawiono ich zastosowania w wybranych projektach naukowych.

Ekstrakcja cech to proces przetwarzania otrzymanych danych, który prowadzi do uzyskania reprezentacji odpowiednio sprofilowanej do analizowanego problemu. Tym samym przyczynia się do poprawy wydajności przetwarzania danych i optymalizacji procesu modelowania oraz umożliwia pozyskiwanie atrybutów, które mogą być wykorzystywane zarówno przez ekspertów dziedzinowych, jak i algorytmy uczenia maszynowego. Wyróżnia się dwie zasadnicze fazy tego procesu: pierwsza to konstrukcja nowych cech (ang. feature engineering), natomiast druga to wybór najistotniejszych spośród uzyskanych w ten sposób atrybutów (ang. feature selection). Istnieje wiele podejść do automatyzacji procesu tworzenia i selekcji atrybutów, trudno jednak znaleźć metody wspierające interakcję z użytkownikami, które uwzględniałyby wiedzę dziedzinową pozyskiwaną od ekspertów, ich doświadczenie i preferencje.

W badaniach nad interaktywnością procesu ekstrakcji cech poruszono problemy związane z uzyskiwaniem użytecznych i zrozumiałych dla ekspertów atrybutów z wielowymiarowych danych, a także możliwość ograniczenia ilości tych atrybutów w celu uzyskania możliwie najprostszych, ale dokładnych modeli. Zaproponowane w rozprawie nowe metody interaktywnej ekstrakcji cech wykraczają poza obecnie znane standardy, umożliwiając skuteczniejszy sposób wyrażania wiedzy dziedzinowej związanej z najważniejszymi podzbiorami atrybutów. Zaproponowane algorytmy konstrukcji i doboru cech wykorzystują różne formy granulacji przestrzeni atrybutów, a także pozwalają na wydajne przetwarzanie dużych danych poprzez zrównoleglenie obliczeń. Na szczególną uwagę zasługuje zaproponowana metoda uodpornienia algorytmów selekcji atrybutów na ewentualne braki w danych, która pozwala znacząco

zmniejszyć wymiarowość danych gwarantując jednocześnie zachowanie niezbędnego poziomu informacji (wg zadanego kryterium) do predykcji zmiennej celu, nawet po usunięciu określonej liczby atrybutów.

Przedstawione podejścia do ekstrakcji cech zostały wypracowane na podstawie doświadczeń z projektów naukowych z dziedziny analizy danych tekstowych oraz przetwarzania strumieni sensorycznych. Przedstawione metody zostały zweryfikowane pod względem jakości uzyskanych cech, jak również przepustowości, skalowalności i stabilności działania. Zaproponowane rozwiązania zostały zweryfikowane w ramach miedzynarodowych konkursów analizy danych.

Contents

Chapter 1

Introduction

Every day, the surrounding world is monitored by a still increasing number of sensors. Starting with commonly available sensors from our neighborhood, like mobile phones, automotive sensors, wearables, smart home appliances [17, 200] through medical and telemedical devices [2] for respiratory monitoring [332], auscultation analysis [132], cancer diagnostics [326], or rehabilitation support [201], ending with sensors deployed in factories or coal mines [179, 356] to support diagnostics of manufacturing processes and human staff safety assurance. The variety, variability, and velocity of data have therefore arisen, putting additional pressure on data analysis tools and methods. On the one hand, they should provide the possibility to process various types of data in a multitude of very specialized domains of application. On the other hand, they should seamlessly adapt to drifts, shifts, or the emergence of previously unobserved concepts in data, by interaction with domain experts and data scientists [177].

Interactive data exploration techniques allow analysts to discover interesting dependencies in data due to a fact that it gives the ability to efficiently verify current hypotheses about investigated phenomena and formulate new ones. In practice, this is usually done by conducting various tests on available data and using the results of those tests in consecutive stages of the data exploration process. Very often, the main objective of an analyst is to define such a representation of objects described in the data, that in the future will be the most useful for, e.g., constructing prediction models. There are plenty of methods for automatic feature extraction that are well-described in literature [149, 165, 199, 242, 280]. However, referring to *Judea Pearl*, two more ingredients are needed to move from traditional statistical analysis to causal inference, namely: *"a science-friendly language for articulating causal knowledge, and a mathematical machinery for processing that knowledge, combining it with data and drawing new causal conclusions about a phenomenon"* [290]. The need to embed domain knowledge in data has already been recognized by many scientists as one of the most challenging areas of research [26, 103, 293].

The essential difference between human perceptions and machine-generated measurements, pointed by *Lotfi Zadeh*, is that *"measurements are crisp whereas perceptions are fuzzy"* [417]. Indeed, machine-generated data are often very vague to users. Endless chains of numbers generated by sensor networks are not even close to real-world concepts, problems, and entities [184, 260]. Therefore are particularly hard to interpret. One of the challenges, pointed by *Leslie Valiant, is "to characterize*

the computational building blocks that are necessary for cognition" [350]. In that context, it is essential to describe the data in a possibly intuitive way. The idea is to make an intelligent use of the information granulation paradigm in the context of aggregating, selecting, and engineering attributes (features/variables/dimensions) that describe the data [139, 415]. For example, we may refer to such straightforward techniques as statistics characterizing granules over sliding time windows. In the case of the underground coal mine sensors, derivation of multivariate series of window-based statistics allows data analysts and experts to deal with noisy and incomplete data sources, better reflect temporal drifts and correlations and reliably describe real situations using higher-level data characteristics [356]. Extracting meaningful features is important in many domains, like medicine or criminal justice [119, 325]. Combining machine-generated data with attributes corresponding to experts' assessments proved to have a positive impact on the quality and robustness of the machine learning models, e.g., in the case of seismic bumps prediction [179].

Feature engineering is recognized as an important but laborious approach [28]. As opposed to the above, representation learning opts to augment artificial intelligence with the capability to autonomously (i.e., without human interaction) identify and disentangle the underlying explanatory factors hidden in the low-level sensory data. The concepts of deep, distributed representations and unsupervised pre-training have recently become a dynamically developing area of research with many successful applications [131, 284, 309, 382]. There are many methods allowing to project or embed the data onto new derived dimensions, forming so-called latent concepts – typically, a combination of (almost) all attributes of the original data [151]. However, regardless of the achievements in the areas, these algorithms generally suffer from the lack of interpretation of the projected dimensions [164, 325] and for that reason are not investigated further in this study. In contrary to feature engineering, which though often labor-intensive, is a way to take advantage of human ingenuity and prior knowledge. Thereby, justifying the effort put into the design of data transformations and preprocessing pipelines when deploying machine learning algorithms.

Let us also emphasize the problems associated with the increasing dimensionality of data, which may exceed human perception. In such cases, users may be still able to interpret attributes' meaning but navigation through their subsets becomes harder. The curse of dimensionality is a well-known problem to machine learning methods, as well. To address this aspect of complexity, one can operate with clusters of attributes inducing similar information (e.g., similar partitions) or employ some techniques of attribute selection, which would replace large attribute sets with their minimal subsets providing comparable information about data [182]. Searching for such subsets is a well-established task within the theory of rough sets (RST) [370]. Given an initial set of attributes, one can search for so called reducts, which induce (almost) the same information as all considered attributes [285]. A number of heuristic methods have been developed to derive the most interesting reducts from large and complex data tables [72, 285].

The task of feature selection may be defined at two levels. Predicting near-future readings of a particular sensor, one could think about it in terms of choosing other relevant sensors and time frames that are enough to start the process of model training. In this case, it is crucial to interact with domain experts and provide

appropriate analytic reports/visualizations to let them make the right assessments. This level might be also called an information source selection. The second level considered in the thesis refers to selecting specific features constructed during the sliding time windowing process. The reference to the sliding time window technique is an example of a situation where at a higher level of granularity, essential characteristics need to be defined. Indeed, from the user's point of view, operating on statistics extracted from numerical series covered by a given window interval is definitely more natural than operating on raw numerical data. At this level, one can successfully proceed with relatively simple methods, which yield surprisingly small feature sets in order to establish an efficient framework for deploying, monitoring and tuning the forecasting models embedded into the production system.

In some applications, e.g., related to sensor-based hazard monitoring or medical diagnostics, besides the accuracy, speed, and reliability of a prediction model, no less important is resilience. A single sensor failure (or interruption of signal transmission), which typically causes a missing whole dimension of data, cannot result in the inability to assess the situation. To address this issue, we may refer to the broad studies on missing data imputation techniques [46, 258]. These are often based on univariate series analysis or sampling from original data distribution and may have problems dealing with longer gaps, e.g., resulting in a higher level of uncertainty for subsequent assessment [338, 362]. As an alternative, some researchers study non-imputation methods designed for the classification or regression of incomplete data. Such methods may rely on aggregation techniques and higher-level features (or granules of features) that are less sensitive to missing data [143, 396], ensembles of diverse classification of regression models [179, 395], or enhanced predictive models with additional (redundant) checks, e.g., verifying cuts in decision trees [27].

The aforementioned approaches rely on the assumption that features selected for the process of model learning do contain additional (redundant) knowledge, whereas state-of-the-art feature selection techniques attempt to remove redundancy. Our goal is to formulate new constraints, whereby selected feature sets are guaranteed to provide enough information about the considered target variables even if some of those features are dropped. One of the discussed approaches is to rely on a collection of diverse feature subsets with their corresponding prediction models treated as an ensemble. Another approach is to search for feature sets with a guarantee of providing sufficient predictive power even if some of their elements are missing. In the dissertation, we introduce the idea of resilient feature selection. In particular, we formulate the rough-set-based notion of r-$\mathbb{C}$-reducts – an irreducible subset of features providing a satisfactory level of information about the considered target variable even if up to r features are unavailable.

In the study on the interactiveness of the feature extraction methodologies, we address the problems of deriving useful and understandable parameters (attributes, features) from raw sensor readings and reducing the amount of those parameters in order to achieve possibly simple yet accurate models. In the dissertation, a number of innovative approaches to automating feature creation and selection are proposed. The current state of knowledge on feature extraction processes used in commercial applications is shown. The problems associated with processing big data sets and approaches to process high-dimensional time series derived from sensor networks

are discussed. Although we rely mainly on RST to specify requirements related to the design and implementation of our approach to interactive attribute selection, the framework presented in this paper can be used together with other well-known methodologies of data analysis.

1.1 Plan of the Dissertation

In Chapter 2, an overview of the state-of-the-art feature extraction methods is presented. In particular, in Section 2.1 a review of feature engineering techniques is provided, with a special emphasis put on sliding window techniques to feature creation. To provide a proper context of other related approaches that are not covered in this study, in Section 2.2, we present a comprehensive review of representation learning and dimensionality reduction methods. In Section 2.3, we recall relevant approaches to feature selection. In Section 2.4, we discuss various approaches to information granulation in feature extraction. In Section 3.1, we recall basic concepts from the theory of rough sets (RST).

In Chapter 3, we introduce the idea of resilient feature selection and, accordingly, we introduce r-$\mathbb{C}$-reducts. In Section 3.1, we introduce the notion of criterion function $\mathbb{C}$, which enables us to consider various feature selection formulations at a higher level of abstraction. In particular, we show how $\mathbb{C}$ generalizes the RST-based feature selection approaches relying on various definitions of reducts and approximate reducts. In Section 3.3, we outline an Apriori-inspired algorithm that generates all r-$\mathbb{C}$-reducts of a given type. In Section 3.4, we study the tasks of resilient feature selection from the perspective of their computational complexity. We prove that many NP-hard feature selection / elimination problems remain NP-hard for any arbitrary resilience level r. In Section 3.5, we present heuristic DFS algorithms for searching for optimal r-$\mathbb{C}$-reducts, with specific examples of permutation-based and approximation methods.

In Chapter 4, we outline our approach for feature extraction, aimed at processing multivariate time series. In Section 4.1.2, we describe the data in a possibly intuitive way, using statistics characterizing sliding time windows. In Section 4.2 and its Subsections: 4.2.1, 4.2.2, and 4.2.3, we discuss how the concept of granulation can be made useful in selecting and engineering features on large and, possibly, complex data sets. Finally, in Section 4.3, the complete framework for linking resilient feature selection and machine learning techniques to build a predictive model resistant to partial data loss is proposed.

In Chapter 5, we provide a comprehensive experimental evaluation of the introduced feature extraction methods over large, multivariate time-series data across significantly different domains. The analysis of potentially dangerous methane concentration and seismic events are presented in Section 5.1 and 5.2, respectively. In Section 5.3, we evaluate the performance of the introduced framework in the fire and rescue domain that refers to the analysis of data collected from body sensor networks. In this Chapter, we also evaluate the impact of the developed feature extraction methods on the prediction quality and resilience of various machine learning models. As an important argument for considering interactive feature extraction processes and built-in human-computer interaction into machine learning processes, let us stress out that in the case of seismic data, training models on both sensor readings and domain

experts' assessments allowed us to improve the quality of the predictors significantly. In Section 5.4, we performed short-term spot prices prediction of many univariate time series collected from the AWS Cloud Spot market. Furthermore, we describe a series of international data mining challenges organized to facilitate this study.

In Chapter 6, we conclude the dissertation and we elaborate on some future research directions.

1.2 Main Contributions

The main contributions of this dissertation are:

In Chapter 2, a broad overview of state-of-the-art feature extraction techniques is provided, with a special emphasis put on sliding window techniques to feature creation and rough set-based granulation and feature selection techniques. Namely, to various extensions of reducts (Definition 1).

In Chapter 3, we introduce a new idea of resilient feature selection – based on so-called r-$\mathbb{C}$-reducts – attribute sets that are well-suited for the investigated problems and provide a level of redundancy that makes these sets more invulnerable with respect to possibly missing or questionable attribute values. The introduced r-$\mathbb{C}$-reducts extend the classical notion of a reduct developed within the rough set theory (which is briefly discussed in Chapter 2). In the provided notion, $\mathbb{C}$ refers to a function encoding the criterion of preserving enough information by the considered sets of attributes. At the same time, r stands for the number of attributes that can be removed from those sets without making them insufficient to build decision models [137] (at the accuracy level corresponding to $\mathbb{C}$). Functions $\mathbb{C}$ (Definition 5) actually enable us to express a number of so-called approximate attribute reduction criteria known from the RST-related literature, based on, e.g., discernibility, entropy, or positive regions. Consequently, by defining the resilience factor r as combinable with an arbitrary $\mathbb{C}$, we generalize all those formulations. The discussed idea of resilience is surely more general, and one may consider it an extension of many other, not necessarily rough-set-based feature selection methods.

The important theoretical contribution in Chapter 3 refers to a broad discussion on the impact of the resilience on the overall complexity of feature selection problems and algorithms. In particular, in Section 3.1, we generalize the way of reasoning about attribute subsets by introducing criterion functions, which, for each given decision table $\mathbb{S} = (U, A \cup \{d\})$, return a binary assessment of the candidate attribute subsets. We further use criterion functions to provide a generalization of rough set reducts as *criterion reducts* $\mathbb{C}(R)$ (Definition 6). Lastly, we show some theoretical properties of criterion reducts and we define criterion reducts for a number of well known notions of reducts, i.e., approximate entropy reducts ($\mathbb{C}^{(H,\varepsilon)}$-reducts), γ-reducts ($\mathbb{C}^{\gamma}$-reducts), etc. In Section 3.3.2, we prove that any NP-hard feature selection problem understood as the task of finding – for an input decision table – the minimal $\mathbb{C}$-reduct that may be expressed via so-called monotonic criterion functions $\mathbb{C}$ (e.g., the minimal (H, ε)-approximate reduct problem [352], the minimal (γ, ε)-approximate reduct problem [367], a wide family of discernibility-based approximate/partial reduct

optimization problems [270,276], etc.) retains its NP-hardness for arbitrary resilience level r (Theorem 1).

In order to prove NP-hardness – by providing a polynomial time transformation – in Subsection 3.4.1, we introduce a family of artificial attributes, so-called A-attributes, denoted as $\#attr$. In a number of lemmas in Subsection 3.4.1, we show some important properties of A-attributes. Among others, we prove that any r A-attributes $\{\#attr_1, \ldots, \#attr_r\}$ form the smallest r-$\mathbb{C}$-reduct (Lemma 3). We show, in Lemma 4, how to construct $\mathbb{C}$-reduct by adding A-attributes to reduct, and we discuss the impact on $\mathbb{S}$ and $\mathbb{C}$ when data representation is extended with A-attributes (Lemma 5). Our study includes also a visual interpretation of the NPH proof (Section 3.4.3), a broad discussion on the meaningfulness of the provided NPH study and the complexity result derivable directly from Theorem 1 (Section 3.4.4). In particular, referring to Theorem 1, we prove NP-hardness of the resilient version of the minimal (H, ε)-approximate reduct problem, and the resilient versions of the minimal α-reduct and (γ, ε)-approximate reduct problems. Let us note that the same mechanism could be easily applied for many other cases as well, in particular, for any formulations of $\mathbb{C}^{(Q,\varepsilon)}$-reducts for which the corresponding measures Q satisfy conditions of Definition 7 [351,353].

In Sections 3.3 and 3.5, we discuss opportunities of exhaustive and heuristic search of feature subsets, and we assess computational complexity of proposed algorithms. We elaborate on two generic strategies that follow a popular idea of dynamic exploration of the lattice of feature subsets (i.e., Figure 2.2). Namely, breadth first search (BFS) and depth first search (DFS). For BFS, we adapt the well-known Apriori algorithm [331] for the purpose of r-$\mathbb{C}$-reduct search (Section 3.3). In Section 3.5, we consider two approaches to the depth first search exploration of the lattice, which allow us to identify subsets of attributes that satisfy the resilient version of $test_{\mathbb{C}}$ function: $r\text{-}test_{\mathbb{C}}$ (Algorithm 3). Algorithm $r\text{-}test_{\mathbb{C}}$ verifies if a given set of attributes $R \subseteq A$ satisfies the resilient criterion r-$\mathbb{C}$ under the condition that implementation of $test_{\mathbb{C}}$ is given. In Subsection 3.5.1, we present a novel Algorithm 4 generating r-$\mathbb{C}$-reducts inspired with a permutation-based technique that is common for RST-based approaches [353,367]. In Subsection 3.5.2, we discuss the approximation of the permutation-based algorithm for resilient feature selection (Algorithm 5).

In Chapter 4, we outline our approach to feature extraction, aimed at processing data obtained from sensors that provide outputs in the form of time series. In knowledge-based systems, it is common to deploy a potentially large collection of sensors of different types to monitor the environment state and its changes. In such a setting, the gathered data elements can be complex on various levels. Individual readings may take different forms according to the application domain. Values may express continuous phenomena, such as pressure, temperature, or humidity. They can also express a discrete state of the environment, such as an on/off state of a device. Often, data interpretation is possible only in the context of additional knowledge obtained from domain experts. Concerning feature engineering, in Section 4.1.2, we attempt to describe the data in a possibly intuitive way, using statistics characterizing sliding time windows. The proposed approach focuses on extending

the sliding window construction process by adding a number of designed statistics and enhancing it with some more static attributes reflecting assessments obtained from domain experts. This brings the opportunity to compare the prediction quality of models trained using derived features with the expert-based assessment and makes it possible to use features derived from experts in ML models training. In the case of the underground coal mine sensors, derivation of multivariate series of window-based statistics allows us to deal with noisy and incomplete data sources, better reflect temporal drifts and correlations, and reliably describe real situations using higher-level data characteristics, which are common problems in time series analysis, reviewed in Section 4.1.1.

The contributions in Section 4.2.2 refer to a general algorithmic framework for performing feature selection on top of a granular representation of attribute space. Our methodology is devised in such a way that it caters to various types of granules and various goals of feature selection. The purpose is to perform a kind of granular attribute selection that exploits to the fullest semantical relationships between variables. Particular contributions in Section 4.2.1 are concentrated around two aspects. First, we put forward a framework for expressing granules in attribute space. Therein, we include original ideas for discovering and managing similarities between attributes for the purpose of constructing granules. Feature granules can be induced by, e.g., hierarchical clustering on attributes or analysis of so-called *heat maps* that convey the knowledge about attribute interchangeability. On the other hand, we show that meaningful granulations can be derived according to such prerequisites as proximity or common functionality of the considered features.

In Section 4.2.3, we discuss how the concept of granulation can be made useful in selecting and engineering features on large and possibly complex data sets. We show how to utilize the intrinsic properties of the data and underlying problem and background/domain knowledge to build a granular representation of attributes. By taking into account a given granulation of attributes, we can configure our algorithms to achieve faster convergence. The proposed methods are designed in a way to deal with large and complex data sets. We present means to make use of efficient, parallelized computational schemes such as MapReduce. Therefore, the provided tools and examples are devised to work with data sets that are very large in terms of the number of objects and the number and complexity of features. Thus, they address some of the challenges posed by the Big Data paradigm.

As a notable aspect and an important contribution in the frame of this dissertation, let us point out the framework for linking resilient feature selection and machine learning techniques to build a predictive model that is resistant to partial data loss (Section 4.3). In this section, we focus not only on the extracted features and constructed prediction models but also on data processing stages that are designed to let it work within a big data environment and, particularly, with the high dimensional, multi-stream data. In order to provide high-quality assessments, the Algorithm 9 – for blending the models – is designed in a way, which guarantees that a model can be included if it is accurate enough on validation data and sufficiently different from already selected predictors. The diversity may be achieved by employing a variety of models computed on different subsets of attributes and data samples. For this task, the similarity measures (Section 4.2) or resilient attribute subsets (Section 3.2)

may be applied. As a result of blending diverse models, the final ensemble minimizes the impact of concept drifts and achieves a better prediction quality. The results of conducted experiments (Chapter 5) confirmed that the idea is very promising, and resilient learning may significantly minimize the risk and impact of data loss on predictive analysis.

In Chapter 5, we provide a broad experimental evaluation of learning forecasting models over large multi-sensor data sets, including the steps of sliding window-based feature extraction and rough-set-inspired feature subset ensemble selection. We conducted a series of experiments on data connected to the problems of providing safety of miners working underground, which is the fundamental requirement for the coal mining industry. Analysis and proper assessment of potentially dangerous methane concentration (Section 5.1) and seismic events (Section 5.2) significantly improve the safety and reduce the costs of underground coal mining.

One of the considered tasks is to construct a model capable of predicting dangerous concentrations of methane at longwalls of a coal mine basing on multivariate time series of sensor readings. The contributions in Section 5.1 refer to both the analysis of how the nature of sensor readings influenced the architecture of the developed system and the empirical proof that the designed methods for data processing and analytics turned out to be efficient in practice. We show how the complete mechanism can perform on data collected in an active coal mine and processed with the described framework. We show how the complete mechanism can be built into DISESOR - a decision support system in coal mines. The evaluated feature selection approaches yield excellent results even when combined with the simplest possible prediction techniques like logistic regression. Furthermore, we elaborate on the resilience of the solution in the case of partial data loss, e.g., when particular data sources, sensors are damaged or inactive (Subsection 5.1.4).

In Section 5.2, we investigate how the interactive feature extraction and ensemble blending methods, proposed in Chapter 4, generalize to other problems of multi-stream data analysis. Once again, we address the problem of safety monitoring in underground coal mines. This time, we investigate and compare practical methods for the assessment of seismic hazards using analytical models constructed on both raw multi-stream sensory data and features derived from domain experts. The possibility of representing a problem related to data exploration and analysis with machine-generated features, which are additionally enriched with experts' assessments, is one of the essential aspects from the point of view of interactiveness. Furthermore, in Section 5.3, we describe an international data mining challenge organized to facilitate this study. We also demonstrate that the technique used to construct an ensemble of regression models outperformed other approaches used by participants of the described challenge. In Section 5.2.4, we explain how post-competition data was utilized for the purpose of research on the cold start problem in the deployment of decision support systems at new mining sites.

To thoroughly assess the versatility of the developed framework across significantly different domains of application, besides analysis of coal mining-related problems, in Section 5.3, we evaluate its performance in the fire and rescue domain that refers to the analysis of data collected from body sensor networks. The aim of this study is to

assess how automatic feature extraction and classifier learning (without parameters tuning) can cope with the multi-target learning problem. Furthermore, in Section 5.4, we show that, by analyzing spot instance price history and using ARIMA models, it is feasible to perform future spot prices prediction of many univariate time series. The main reason behind the evaluation of ARIMA models on data represented as candlesticks is that both techniques are easy to interpret. Results confirm the quality of the solution, its computational performance, and the versatility of the developed framework resulting in the very short time needed for its adaptation to the significantly different domains.

Some of the partial results of this dissertation were presented at international conferences and workshops. Some were published in conference proceedings and respectable journals. For example, the publications related to the granular and resilient feature selection [137, 139, 142]. Moreover, the research on various applications of feature extraction to improve prediction quality in the field of sensor data analysis in hard coal mining and emergency / firefighting domains [138, 143, 144, 179, 181]. The research on intelligent systems, data models, and processing optimization for interactive feature extraction and data analysis [136, 141, 145, 215, 356–358, 420] are also partial contributions in this dissertation. Some partial results were also published in technical papers and monographs in Polish [140, 178].

1.3 Acknowledgements

I would like to thank my advisors Dominik Ślęzak and Andrzej Janusz for their guidance and invaluable support in writing this dissertation.

I would also like to sincerely thank all from the faculty of Mathematics, Informatics, and Mechanics, University of Warsaw, who I consider not only my colleagues but also my friends. I have learnt a lot from their knowledge and experience. Moreover, I would like to thank all my co-authors: Cas Apanowicz, Ace Dimitrievski, Łukasz Grad, Andrzej Janusz, Mateusz Kalisch, Michał Kijowski, Marcin Kowalski, Michał Kozielski, Zdzisław Krzystanek, Petre Lameski, Marcin Michalak, Sinh Hoa Nguyen, Przemysław Wiktor Pardel, Marek Sikora, Sebastian Stawicki, Krzysztof Stencel, Marcin Szczuka, Dominik Ślęzak, Piotr Wojtas, Łukasz Wróbel, Eftim Zdravevski, whose excellent ideas have taught me to stay open-minded, and have been a motivation and inspiration for my research and writing.

My deepest gratitude goes to my Lovely Wife and Kids for their support in all my efforts and understanding during many years of research activities. None of the things that I achieved would be possible without you. Thank You!

Partial contributions of the dissertation were influenced by the author's work on the following research projects: In grant PBS2/B9/20/2013, new approaches for learning forecasting models from multi-sensor data for the purposes of monitoring natural hazards and industrial processes were investigated . Among others, the research comprised exploring streams of sensor readings registered in underground coal mines and feature engineering that can derive the most meaningful statistics describing multivariate sliding time windows. Grant SYNAT tasks B13 and B14 No. SP/I/1/77065/10 *supported by the Polish National Centre for Research and Development (NCBiR) in the frame of the strategic scientific research and experimental development program:*

"Interdisciplinary System for Interactive Scientific and Scientific-Technical Information". SYNAT project was concerning the problem of creation a universal, open platform for hosting of educational and scientific resources. With a particular focus on: The logical and physical model of data that will be stored in a data warehouse. In implementing Business Intelligence and advanced methods of data analysis based on SQL and data mining, which allow to accelerate the extraction of information. Research on methods of semantic indexing and semantic search of digital objects using dictionaries, thesauruses and ontologies. POIR.01.02.00-00-0150/16 - *conducting research and development of an external, modular tool for the implementation and optimization of artificial intelligence in video games. Research and development project* RPMA.01.02.00-14-7532/17 *in which the problem of application machine learning algorithms to offer products and services on automotive aftermarket was investigated. Research project* 2018/31/N/ST6/00610 *on theoretical foundations of resilient feature selection methods that ensure predictive model operation in case of partial data loss.*

Chapter 2

Feature Extraction

Having in mind the observed variety of possible data representation formats, including text, audio, image, video, relational data, spatio-temporal time series, and many others, it is straightforward that the application of machine learning algorithms and techniques requires a more or less extensive phase of data preparation. Feature extraction (FE) addresses the problem of finding the most compact and informative data representation to improve the efficiency of data storage and processing. The process starts from an initial set of measured data and builds derived features intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and leading to better human interpretations. Logically, there are two phases of this process: the first is the construction of new attributes based on original data (sometimes referred to as feature engineering), the second step is a selection of the most important among the attributes (sometimes referred to as feature selection). In this chapter, we provide a broad overview of the state-of-the-art feature extraction methods, including feature construction, selection, granulation, and selected methods from rough set theory. We also briefly present some other related topics, like dimensionality reduction and representation learning.

2.1 Feature Engineering

Feature engineering (FE) is the process of using domain knowledge of the data to create features that make machine learning algorithms work [199]. The importance of feature engineering was aptly identified by *Pedro Domingos: "At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used"* [93]. Indeed, this process is fundamental to the application of machine learning resulting in simpler and more effective predictive models, improved models' robustness and resilience, reduced computation time and resources needed, and foremost, better interpretability of the results. In this section, we present a comprehensive review of the state-of-the-art in the area with a strong emphasis on the structural, relational, and time series data extraction methods.

Some machine learning algorithms, like a decision tree model, can handle various data representations. However, most have fairly restrictive limitations and usually require a specific data format. For example, rough set reudcts construction methods

can work with categorical data only. In the case of numerical attributes, those algorithms require a discretization to transform continuous attributes before the data reduction may be performed. On the other hand, some algorithms, like neural networks, require numeric inputs [373]. The typical data preprocessing steps that allow adapting attributes' format to the requirements of selected learning methods include: normalization, standardization, discretization, categorical encoding, imputation of missing values, outliers detection, and user-defined custom transformations, e.g., min/max values, percentiles, or generating polynomial features [112, 144].

There are several data preprocessing techniques to be used for encoding categorical variables [302]. One hot encoding is the most widely used scheme to transform a single variable with 'd' distinct values to 'd' binary variables indicating the presence (1) or absence (0) of the particular category. In ordinal coding, an integer value is assigned to each category (assuming that the number of categories is known). Polynomial coding is a form of trend analysis that looks for trends in the categorical variable. Leave-one-out is an example of the target-based encoder that calculates mean target of a given category for each observation, supposing that this observation is removed from the data set. We may also mention sum coding, Helmert and James-Stein encoders, etc. [302]. All leading to converting categorical features to binary, integer, or continuous ones, as expected by ML algorithms' inputs.

In the case of some machine learning algorithms, objective functions may not work at all, or may perform less effectively, without proper feature scaling [10]. For example, we may recall stochastic gradient descent and its variants, which are recognized as an effective way of training deep networks [170, 369]. The need for normalization and standardization arises naturally when dealing with clustering [173], in the case of experiments involving multiple arrays [38], or whenever data are collected from various sources [356]. In some applications of high-density oligonucleotide arrays, the goal is to learn how RNA populations differ in expression in response to genetic and environmental differences. For example, large expression of a particular gene or genes may cause an illness resulting in variation between diseased and normal tissue. The obscuring sources of variation can have many different effects on data. Unless arrays are appropriately normalized, comparing data from different arrays can lead to misleading results [171]. Let us now briefly recall some common approaches to feature scaling.

Given a lower bound $min(a)$ and an upper bound $max(a)$ for an attribute "a", the min-max normalization is one of the elementary methods to scale the range in $[0, 1]$.

$$\hat{a} = \frac{a - min(a)}{max(a) - min(a)}$$

The general formula to rescale a range between values $[\hat{a}^{inf}, \hat{a}^{sup}]$ is given as:

$$\hat{a} = \hat{a}^{inf} + \frac{(a - min(a)) * (\hat{a}^{sup} - \hat{a}^{inf})}{max(a) - min(a)}$$

Mean normalization refers to the average $avg(a)$ values of a feature "a":

$$\hat{a} = \frac{a - avg(a)}{max(a) - min(a)}$$

Standardization (or Z-score normalization) is a technique used to scale the data such that the mean of the data becomes zero and the standard deviation becomes one. Here the values are not restricted to a particular range. We can use standardization in the case of large differences between input data attributes' ranges. Standardization is widely used in many machine learning algorithms, e.g., support vector machines, logistic regression, or deep learning [250]. The general method of calculation is to determine the distribution mean μ and standard deviation σ for each feature a and to replace it with the following formula:

$$\Phi^{\mathbb{M}}_{candidate} = \frac{a - \mu}{\sigma}$$

Furthermore, we may scale features according to a given norm $\|.\|$, i.e., Euclidean length, L_1 (city-block length), or any other user-defined norm. We may also mention rank and quantile normalizations, and their applications in the image processing and genetics [10, 38]. There are also decoupling and Gaussian normalization that are successfully applied in collaborative filtering [190, 191]. In [225] was proposed an interesting framework to handle some special cases when standard normalization techniques are not capable of eliminating technical bias due to skewed distribution of variables. We may also recall a variety of methods adjusted to the given feature distribution [10], which obviously do not close the range.

Many algorithms, like Apriori or Naive Bayes, can handle only nominal or discrete attributes [409]. Even in the case of algorithms, which are able to deal with continuous attributes, learning is far less efficient and effective. Thus an embedded or an external discretization of data is often required [401]. The main goal of discretization is to transform a set of continuous attributes into discrete ones, e.g., by associating categorical values to intervals and thus transforming quantitative data into qualitative data [410]. In this manner, symbolic data mining algorithms can be applied over continuous data, and the representation of information is more concise and specific.

Assuming that the data is represented by a set of objects (instances, observations) U, set of attributes (features, variables) A, and (in the case of supervised problems) a set of classes D, a discretization algorithm would split the continuous attribute $a \in A$ in this data set into k discrete, non-overlapping intervals:

$$A_{discr} = \{[a_0, a_1], (a_1, a_2], .., (a_{k-1}, a_k]\}$$

,where $a_0 = min(a)$ is the minimal observed value of the attribute a (or $-\infty$, if attribute values are not bounded), $a_k = max(a)$ is the maximal value (constant or ∞), and $\forall_{0 \le i \le j \le k}\ a_i \le a_j$. Such a discrete result A_{discr} is called a discretization scheme on attribute $a \in A$, and $A_{cuts} = \{a_1, .., a_{k-1}\}$ is the set of cut points of attribute $a \in A$. Let us briefly present selected, common discretization methods.

Equal-frequency (EFB), equal-width (EWB), and fixed-frequency (FFB) binnings are the simplest and most straightforward discretization methods. All those methods involve sorting of the observed values V_a of a continuous feature a. For the given k number of intervals, EFB divides the sorted values into k intervals, so that each

contains approximately the same number of training instances [97]. Let U refer to the set of observations (objects), then each interval contains $\frac{|U|}{k}$ training instances with adjacent values. Note that training instances with identical values are placed in the same interval, thus it is not always possible to generate k equal frequency intervals.

If a continuous variable is observed to have values bounded by $min(a)$ and $max(a)$, EWB aims to divide the range of observed values into k equally sized intervals (bins), where k is a given constant parameter. The width δ_a of each interval is computed as:

$$\delta_a = \frac{max(a) - min(a)}{k}$$

the cuts (boundaries of the intervals) are defined as: $min(a) + i * \delta_a$, where $i = 1, .., k - 1$.

For a predefined interval frequency k, FFB discretizes the sorted values into intervals so that each interval has approximately the same number k of training instances with adjacent values [410]. All above mentioned methods are applied to each continuous feature independently, hence all are classified as univariate. They also make no use of class information (unsupervised).

The scientific literature provides numerous proposals of discretization techniques, and there are many different axes by which they can be classified, e.g., univariate vs. multivariate, supervised vs. unsupervised, global vs. local, static vs. dynamic, etc. [120]. The most common evaluation measures used by the discretizer to assess the best discretization scheme are derived from information theory (Gini index, entropy), statistics (χ^2, ChiMerge), or Rough Sets Theory (RST) [108, 278, 378]. Furthermore, some methods utilize wrapper approach, like ID3 [308], Bayesian approach [42], fuzzy functions [322], and many other techniques [33, 269]. It is also important to stress out that obtaining the optimal discretization is a NP-complete problem [66].

When analyzing real data sets, one may face a broad spectrum of problems related to data, varying around: missing values, anomalies, exceptions, discordant observations, or contaminants [56, 58]. Missing values imputation has been studied for several decades being the basic solution for incomplete data problems, specifically those where some data samples contain one or more missing attribute values [239, 428]. Outlier detection techniques strive to solve the problem of discovering patterns that do not fit to expected behaviors [389]. This is a particularly challenging and important problem in the case of big sensor networks and multidimensional time series data analysis [211, 356]. The problems related to missing attributes, noisy data, or outliers refer more to data quality aspects and data cleaning rather than to feature engineering. Therefore, some selected approaches to imputation of missing values and outlier detection methods are discussed concisely further in Section 4.1.1.

The widespread growth of Big Data and the evolution of Internet of Things enable various entities to continuously generate and collect streams of data [274]. Stream data analysis is essential for many fields of application where processes are typically monitored by a number of sensor devices [200], such as: logistics, mining industry, health-care, medicine, and even agriculture [73, 379, 423], Proper understanding of data collected from many sensors and application of machine learning methods are very challenging and time-consuming tasks that usually require particular feature engineering [6, 115]. Feature extraction approaches, which output interpretable and

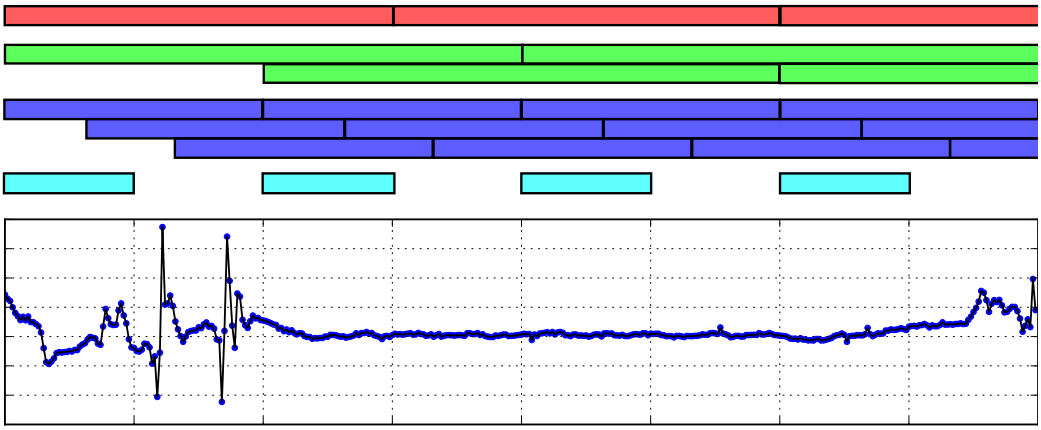


Figure 2.1: A set of possible sliding windows set-ups.

dimensionally consistent features, are still in big demand and are considered as an important research topic [73]. Among them, techniques based on sliding window segmentation are considered as one of the simplest yet very effective for constructing easily interpretable features from time series derived from data streams [356, 423].

Deriving statistics from sliding time windows can be regarded as a crucial FE stage in all knowledge discovery process investigating sensor readings and (multivariate) time series [143, 235]. A sliding window is defined by a *length* and an *offset*. The length determines the size of a window, whether it is a fixed number of readings contained in a window or a fixed time interval. The offset is the extent to which the consecutive windows overlap to each other. In Figure 2.1, we provide four examples of possible sliding window set-ups. The example marked in red shows the situation when the length of a sliding window is equal to the offset. The green and blue examples show the consecutive positions of a sliding window when the offset is equal to $\frac{1}{2}$ and $\frac{1}{3}$ of the length. The example marked in cyan illustrates the situation when the offset is twice as large as the length (or in general just greater) of a sliding window [144].

Sliding time windows are represented by various statistics computed over their values [138]. In practice, such methods require an extensive feature engineering step, which often needs to be domain-specific [44, 419]. For a comprehensive study on efficient maintenance of basic statistics derived from sliding windows, we may refer to [82]. For an example how to integrate multi-sensor analysis with external sources of spatio-temporal information, let us refer to [328]. An example of utilization of such statistics as higher-level features can be found in [394]. Time series can be also filtered or smoothed (using, e.g., running averages) in order to reduce its complexity while maintaining its important characteristics [235]. Series of data points can be approximated using methods, such as: piecewise constant approximation [208], or piecewise linear representation [366]. Selection of an appropriate time series representation is the fundamental aspect of an efficient analysis of sequential data. For a more detailed overview of approaches to time series representation, one may refer to [115].

In a broader sense, algorithms and systems for the on-line prediction based on sensor readings can be placed within the scope of research on time series data mining [296], or pattern recognition from multivariate time series [113, 301]. In

this field, a lot of research was conducted on topics such as searching for similar subsequences [81,212], or time series segmentation and dimensionality reduction [207]. In tasks such as subsequence matching, a sliding time window approach is used in a combination with series compression techniques, e.g., the symbolic aggregate approximation [238]. In many domains, series transformations such as the discrete Fourier transforms (DFT) are also often applied in this context [418]. Obviously, sliding time windows approaches may vary depending on the area of application [251]. Nevertheless, the overall mechanism of computing time-window-based representations can be treated as a universal approach.

There are numerous well-known automatic feature engineering methods. In some approaches, they are tightly integrated with the modeling process, e.g., hidden layers of a deep neural network model internal representations in a way analogous to constructed features. In other approaches, they are limited to simple preprocessing of data. Still, extracting meaningful features that describe the studied problem at a higher level of abstraction, e.g., by a proper data granulation, thus allowing easier interpretation of the predictive models' outcomes, is considered a very challenging task, important in many domains [119,325]. One of the possible approaches to feature engineering, which is sometimes required to convert "raw" data into a set of useful and meaningful attributes, is related to human expertise and creation of manually crafted data extractors and transformations. Despite the evident value of the features obtained this way, leading to easily interpretable and well suited data representation, in some cases this method may be far too expensive and time-consuming. Therefore, by complementing it with automatic methods, one can achieve a viable compromise between the possibility to process big volumes of data and taking advantage of human expertise. With this respect, we may refer to already mentioned statistics characterizing granules over sliding time windows, which may be easily defined or interpreted by users. Furthermore, as discussed later in the dissertation, the process of sliding window-based feature creation may be automated, and the derived data representation may be complemented by experts' assessments.

2.2 Representation Learning and Dimensionality Reduction

Representation learning (or feature learning) allows to automatically discover the representations from raw data. This approach is an established alternative to classical feature engineering. There are many feature learning methods that can be either supervised (e.g., neural networks) or unsupervised (e.g., matrix factorization, auto-encoders), linear (e.g., linear discriminant analysis) or nonlinear (e.g., kernel methods). However as the number of features increases, the model training takes far more time, and consumes more compute resources and storage. Trained predictors may become more complex, and may relay on misleading, redundant, or noisy information. This may lead to decreased models' accuracy and over-fitting. There are many methods allowing to project or embed the data into a lower dimensional space while retaining as much information as possible. Classical examples are singular value decomposition, principal component analysis, kernel principal

component analysis, independent component analysis, multidimensional scaling, word embeddings, auto-encoders, deep learning, etc. [50, 384]. In this section, we provide an overview of the state-of-the-art in the area.

Singular value decomposition (SVD) and principal component analysis (PCA) are two commonly used dimensionality reduction methods that attempt to find linear combinations of features in the original high dimensional data matrix to construct a meaningful, yet compressed representation of a data set. They are preferred by different fields of application. PCA is often used for bio-medical data, or in genetics [102, 126]. Meanwhile, SVD is more popular when the investigated problem is related to sparse representations, e.g., in (mechanical) faults diagnosis, or in the case of complex chemical processes analysis [158, 407].

SVD is a factorization of a real (or complex[1], i.e., $\mathbb{Z}$) matrix that generalizes the eigen-decomposition of a square matrix (i.e., $n \times n$) to a rectangular one (i.e., $m \times n$). More formally, with SVD any real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$:

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

is decomposed into the product of two unitary[2] matrices $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V}^T \in \mathbb{R}^{n \times n}$, and a diagonal rectangular matrix of singular values $\mathbf{\Sigma} \in \mathbb{R}^{n \times m}$. The general formula $\mathbf{A} = \mathbf{U} \times \mathbf{\Sigma} \times \mathbf{V}^T$ is shown in-detail below:

$$\mathbf{A} = \begin{pmatrix} \overrightarrow{u_1} & \overrightarrow{u_2} & \cdots & \overrightarrow{u_m} \end{pmatrix} \times \begin{pmatrix} \sigma_1 & & & \\ & \ddots & 0 & \\ & 0 & \sigma_r & \\ & & & 0 \end{pmatrix} \times \begin{pmatrix} \overrightarrow{v_1} & \overrightarrow{v_2} & \cdots & \overrightarrow{v_n} \end{pmatrix}^T$$

where vectors $\overrightarrow{u_i} \in \mathbb{R}^m$, $\overrightarrow{v_i} \in \mathbb{R}^n$, and the singular values σ_i on the diagonal of the matrix $\mathbf{\Sigma}$ are non-negative and ordered according to their importance, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, where $r \leq min(m, n)$ is the rank of the matrix $\mathbf{A}$. Naturally, we may compress all the matrices in the above formula with the rank r of the original matrix $\mathbf{A}$. In such a case: $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$, and $\mathbf{V}^T \in \mathbb{R}^{r \times n}$.

The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set $\mathbf{A} \in \mathbb{R}^{m \times n}$ (consisting of a potentially large number of interrelated variables) retaining as much as possible of the variation present in the data. This is achieved by transforming original data representation to a new set of variables, so-called principal components, which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables. I the first step, we center the data in matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ by subtracting it with matrix $\mathbf{A}_{mean}$ consisting of mean vectors for each column in matrix $\mathbf{A}$:

[1] Later in this section, we discuss real matrices, as they are more relevant for real-life data sets.

[2] We call a matrix $X \in \mathbb{Z}^{n \times n}$ unitary iff $XX^H = X^H X = \mathbb{I}$. For a real matrix $X \in \mathbb{R}^{n \times n}$, we have $X^H = X^T$, and we say that a matrix is orthogonal, i.e., $XX^T = X^T X = \mathbb{I}$

$$\mathbf{A}_{mean} = \begin{pmatrix} mean(a_{1,1},..,a_{m,1}) & \cdots & mean(a_{1,n},..,a_{m,n}) \\ mean(a_{1,1},..,a_{m,1}) & \cdots & mean(a_{1,n},..,a_{m,n}) \\ \vdots & \ddots & \vdots \\ mean(a_{1,1},..,a_{m,1}) & \cdots & mean(a_{1,n},..,a_{m,n}) \end{pmatrix}$$

This way, every column in matrix $\mathbf{B} = \mathbf{A} - \mathbf{A}_{mean}$ has a zero (0) mean. The next step is to calculate the co-variance[3] matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ for the columns (features) in table $\mathbf{B} = \left(\overrightarrow{b_1} \cdots \overrightarrow{b_n}\right)$. Since every column in $\mathbf{B}$ has a zero mean (i.e.,$\forall_{1 \leq i \leq n} E[\overrightarrow{b_i}] = 0$) then co-variance between features:

$$Cov[\overrightarrow{b_x}, \overrightarrow{b_y}] = \frac{1}{m} \sum_{1 \leq i \leq m} (x_i - E[\overrightarrow{b_x}])(y_i - E[\overrightarrow{b_y}]) = \frac{1}{m} \sum_{1 \leq i \leq m} x_i * y_i$$

where x_i, y_i correspond to i-th observations (rows) in $\overrightarrow{b_x}$ and $\overrightarrow{b_y}$, respectively. Hence, we may express a co-variance matrix as follows:

$$\mathbf{C} = \frac{1}{m} \mathbf{B}^T \mathbf{B}$$

Here, we can calculate eigenvectors, and the corresponding eigenvalues, for matrix $\mathbf{C}$, such as:

$$\mathbf{CV} = \mathbf{V\Sigma}$$

where matrix $\mathbf{V}$ contains eigenvectors, a diagonal matrix $\mathbf{\Sigma}$ contains eigenvalues[4]. For the purpose of dimensionality reduction, we can project the data points onto the first k principal components, i.e., truncating matrix $\mathbf{V}$ to only k most significant features ($\mathbf{V}_k$) and projecting the original data $\mathbf{A_k} = \mathbf{AV}_k$ retaining enough variance. The first principal component is the direction in feature space along which projections of observations have the largest variance. The second principal component is the direction which maximizes variance among all directions orthogonal to the first one. The k-th component is the variance-maximizing direction orthogonal to the previous k-1 components.

It is also worth mentioning a few other dimensionality reduction methods [18]. Fisher's linear discriminant allows to find a linear combination of features that separates two or more classes of objects. Linear discriminant analysis is a generalized version of Fisher's linear discriminant, typically used for compressing supervised data [365]. This technique projects data in a way to maximize the target class separability. In independent component analysis, the original inputs are linearly transformed into features which are mutually statistically independent. Robust principal component analysis is proposed since the standard PCA is very sensitive to noise or outliers, and the estimated values obtained by PCA can be arbitrarily far from the true value [247]. Kernel principal component analysis (KPCA) is an extension of conventional PCA that is capable of constructing nonlinear mappings that maximize the variance in the data. Multilinear principal component analysis (MPCA) is a multilinear subspace

[3] $Cov[X,Y] = E[XY] - E[X]E[Y]$, or $Cov[X,Y] = \frac{1}{m} \sum_{1 \leq i \leq m} (x_i - E[X])(y_i - E[Y])$

[4] In literature, matrix $\mathbf{V}$ is often denoted as $\mathbf{W}$, whereas $\mathbf{\Sigma}$ as $\mathbf{\Lambda}$. We, however, continue with the notation as introduces with SVD example above.

learning algorithm. Compared with other commonly used dimensionality reduction algorithms, MPCA has proven performance for the tensor data [147]. Autoencoders are a specific type of neural networks that uses an adaptive encoder to transform high-dimensional data into a low-dimensional code to then reconstruct the output from this representation [161, 244]. In [18, 365, 381], a comprehensive review of more related methods can be found.

Reduction techniques, like PCA, are useful for 2D or 3D visualizations of high-dimensional datasets [432]. Given a matrix $D^{m \times m}$ with distances between each pair of m objects form the original set, and a number of dimensions (typically, 2 or 3, for 2D or 3D output), multidimensional scaling (MDS) places each object into low-dimensional space in a way that preserves (as well as possible) pairwise distances between object. In genetics and microbiology, typical data analysis pipelines include a dimensionality reduction step for visualising the data in two dimensions, frequently performed with t-distributed stochastic neighbour embedding (t-SNE) [213]. A self-organizing map (SOM) is a type of artificial neural network used to produce a low-dimensional, nonlinear approximation of data. This makes it an appealing instrument for visualizing and exploring high-dimensional data, with a wide range of applications [305]. In addition to already mentioned, there are many more methods that can be used for a similar purpose, including: locally linear embeddings (LLE), isomap, or Laplacian eigenmaps [233].

In the case of texts, the raw data, i.e., a sequence of symbols with variable length, cannot be used directly to the already mentioned algorithms as most of them expect numerical feature vectors with a fixed size. Extraction of text features is an important matter for information retrieval (IR) or natural language processing (NLP) [141]. The standard methods derived form IR refer to tokenization, lemmatization, removal of stop words, Tf-Ifd term weighting, or building various n-gram representations for document corpus, etc. [256]. Below, we present 5 exemplary documents $\{D_1, D_2, D_3, D_4, D_5\}$ to better depict some of the reviewed concepts.

D_1: *"Role of granulation in feature selection"*
D_2: *"On resilient feature selection with* r*-*$\mathbb{C}$*-reducts"*
D_3: *"Interactive attribute selection with reducts"*
D_4: *"Predicting seismic events"*
D_5: *"Forecasting seismic events"*

Word embedding is one of the core feature learning techniques in NLP, where documents are mapped to vectors of real numbers [13]. In its simple form, the embedding may be represented as a term-document incidence matrix $M^{m \times n}$, where rows refer to m documents in corpus, columns refer to the n unique terms constituting the vocabulary of the document corpus, and cells $m_{i,j} \in M$ may determine whether i-th document contains j-th term, or a number of times each term occurs per document. There are many variants of this technique, e.g., by combining it with Tf-Ifd or word co-occurrence[5]. In Table 2.1, a simple term-document incidence matrix for exemplary documents $D_1, \ldots, D_5$ is presented.

[5]For a given corpus, the co-occurrence of two words is the number of times they appear together (and are close enough, e.g., no more than 30 words separates them in text) in documents.

Table 2.1: A term-document incidence matrix for the exemplary documents.

$Doc \backslash Term$	attribute	event	feature	forecast	granule	interact	predict	reduct	resilient	role	seismic	select
D_1	0	0	1	0	1	0	0	0	0	1	0	1
D_2	0	0	1	0	0	0	0	1	1	0	0	1
D_3	1	0	0	0	0	1	0	1	0	0	0	1
D_4	0	1	0	0	0	0	1	0	0	0	1	0
D_5	0	1	0	1	0	0	0	0	0	0	1	0

Basing on co-occurrence, we may discover hidden similarities between words. Latent semantic indexing (LSI) relies on SVD to identify relationships between terms and hidden topics[6] contained in text. LSI assumes that words which are close in meaning often occur in a similar context. For example, cosine similarity for vectors representing terms "attribute" and "feature", or "predict" and "forecast" in Table 2.1 would indicate that those terms are dissimilar, whereas LSI would discover their similarity since both appear in a similar contexts. There are many other methods for topic modeling, besides LSI, it is important to mention latent Dirichlet allocation (LDA), which is one of the most popular in this field of study [187]. On the other hand, explicit semantic analysis (ESA) augments text representations with concept-based features, which are automatically extracted from massive human knowledge repositories such as Wikipedia. This way, it is possible to assign a human-readable name for hidden topics, or even to automatically generate a short substitute summary for documents [266]. Global vectors for word representation (GloVe), is a global log-bilinear regression model for the unsupervised learning of word vectors that is also basing on word co-occurrences [295]. GloVe combines the advantages of the two major model families, i.e., global matrix factorization and local context window methods.

Computing distributed word representations using neural networks is yet another very interesting technique because the learned vectors encode many linguistic regularities and patterns [131]. The skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of syntactic and semantic word relationships. The continuous bag of words (CBOW) model attempts to predict the current target word (the center word) basing on its context (surrounding words) [261]. For the exemplary document D_4: *"Predicting seismic events"*, for the context window of size 3, the task would be to predict the central word *"seismic"* having the context words: *"predict"* and *"event"*[7]. In [67], authors observe that – in the case of statistical machine translation – adding features computed by neural networks consistently improves the performance.

Recurrent neural network (RNN), and particularly long short term memory networks (LSTM), form a broad group of architectures that handle sequential data

[6]The main topic for documents $\{D_1, D_2, D_3\}$ could be related to "feature selection".

[7]It is worth mentioning that the neural network input is a numeric vector embedding for each word (typically, word vectorization is performed after the initial preprocessing).

such as natural language, and hence are particularly useful for NLP. Transformers use attention mechanism to gather information about the relevant context of a given word, and to encode that context in the vector representation [382]. Likewise many other techniques, attention mechanism, which was initially invented for machine translation, has found applications in many other tasks, and currently, can help understanding objects' inter-relations in an image just as well as it supports machine translation tasks [284]. Bidirectional encoder representations from transformers (BERT) is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers [88]. Natural language processing comprises a much wider range of diverse tasks, such as: part-of-speech tagging, chunking, named entity recognition, textual entailment, question answering, or semantic role labeling, and is supported by a vast amount of divers representation learning techniques [70, 310, 375].

The recent advances in objects recognition and image classification were achieved mainly due to convolutional neural networks (CNNs) [62, 218]. The term convolution refers to the mathematical combination of two functions. In the case of CNN, convolution is a specialized type of linear operation used for feature extraction, typically represented as $N \times N$ matrix, which is sometimes referred to as kernel, mask, convolution matrix, or filter [373]. It is used to enhance an image representation via blurring, sharpening, embossing, edge detection, etc. A convolution layer is a fundamental component of the CNN architecture that performs feature extraction, which consists of a combination of a convolution operation and an activation function. Typically, CNNs include also pooling layers to reduce dimensions of data and to provide effective controls for over-fitting. Still, automatic learning of high quality features is considered as a challenge also in this field of study [230]. To improve the process of feature engineering from sequential data performed by traditional CNNs, the convolutional recurrent neural network model extracts features from hidden states or outputs of the recurrent layer [209]. Along to their unquestioned role in image classification, CNNs were successfully applied in many other domains, including natural language processing [70], time series analysis and algorithmic financial trading [336], human activity recognition using multiple accelerometer and gyroscope sensors [152], or in radiology where a deep convolutional neural network was designed to detect COVID-19 cases from chest X-ray images [390].

In the case of images, CNNs practically outperformed all other approaches to feature engineering. However for videos, the well crafted features play a major role. There exists a large number of approaches for extracting local spatio-temporal features, including histograms of oriented gradients (HOG), histograms of optical flow (HOF), and combination of those two [226]. Another popular descriptors are: SIFT [333], and motion boundary histograms (MBH), which rely on differential optical flow [77]. Spatio-temporal interest points encode video information at a given location in space and time. In contrast, dense trajectories track a given spatial point over time to capture motion information [388].

Recently, a variety of model designs and methods have blossomed. There is, however, a hidden catch: the reliance of these models on hand-labeled training data. It is easy to collect and store a large amount of data, however it is difficult and time-consuming to label data, since interaction with human experts

is usually essential for this process [177]. Deep hierarchical representations carry some interesting advantages with that respect. On the one hand, they promote the re-use of features, e.g., by unsupervised learning of intermediate representations, which can be used on a variety of supervised learning tasks. On the other hand, deep architectures can potentially lead to more abstract features at higher layers of representations [28]. Shared representations are useful to handle multiple modalities or domains, or to transfer learned knowledge to tasks for which few or no examples are given but a task representation exists. Learning reusable feature representations from large unlabeled data sets has been an area of active research. For example, one way to build good image representations is by training generative adversarial networks (GANs), and later reusing parts of the generator and discriminator networks as feature extractors for supervised tasks [309]. Much research has been dedicated to learning, understanding, and evaluating the representations of both supervised and unsupervised pre-training methods. With that respect, unsupervised multi-task learning is a promising area of research [324]. For example, generative pre-trained transformer trained on general language data sets can be fine-tuned to specific language tasks [310]. There are more areas of data science, which consider similar problems, e.g., transfer learning [311], weak supervised learning [315], or active learning [177].

As discussed in this section, modern approaches to representation learning and deep neural networks (DNN) enable performing feature extraction with various network architectures [28,32]. The feature extraction and selection is often performed as an implicit phase in training of network's hidden layers. We can think of DNNs trained by supervised learning as performing a kind of representation learning. The last layer of the network is typically a linear classifier, such as a softmax. Whereas, the hidden layers of the network learn to provide a representation to this classifier. In many applications, features extracted form hidden layers are processed directly [209], whereas in statistical machine translation this is a natural model behavior [67].

Deep learning methods employ multiple processing layers to learn hierarchical representations of data, and have preeminent results in many domains [88, 373, 390]. Naturally, there are many more topics related to dimensionality reduction and representation learning [57], including: restricted Boltzmann machines, deep belief networks, or graph neural networks, which have shown high capability in handling relational dependencies behind multivariate time series forecasting where variables depend on one another [402]. This section provides only a high level overview of those techniques, which, in many cases, can be considered as a foremost alternative to the feature extraction methods. However, regardless of the unquestioned achievements in this area, these algorithms generally suffer from the lack of interpretation of the projected dimensions [164, 425], and for that reason, they are not studied further in this dissertation.

2.3 Feature Selection

Feature selection (FS), sometimes referred to as variable elimination, or attribute subset selection, is the process of determining those attributes that potentially contribute to the predictive models. Along to dimensionality reduction, discussed

in Section 2.2, FS is one of the most popular approaches defying the curse of dimensionality, by removing irrelevant and redundant attributes from data [59]. There are many benefits of eliminating surplus variables. On the one hand, the excessive amount of features increases the time and compute resources required to train models. On the other hand, training models on a large number of features may lead to over-fitting, resulting in their lower performance. Furthermore, FS is facilitating data visualization, providing a better understanding of the underlying process that generated the data [148]. Feature selection not only simplifies the obtained data representation, but also allows to acquire features that can be easily utilized by both analysts and learning algorithms [194]. FS can be designed at different levels spanning from a standard tabular data scenario, whereby features take a form of the existing columns / attributes, toward determination of data sources that can be used to extract features in further steps [277]. Feature selection mechanisms can be also combined with other approaches to machine learning and knowledge discovery, e.g., by means of analyzing components of neural network structures (interpreted as features) in order to achieve compact hybrid data representations [323]. FS has become increasingly important for data analysis with numerous successful applications in real life machine learning problems in various domains [53, 174, 189].

Due to the large search space, FS is a difficult combinatorial problem, i.e., for a data with n features the number of possible solutions is 2^n [349, 391]. Searching for a (near)optimal subset of features is a challenging optimization problem, for which many meta-heuristics, including: bee or ant colony optimization [123], simulated annealing and whale optimization [252], Harris hawks [159] or grey wolf optimizers [1], have been successfully applied. We may also distinguish several search strategies to select a subset of variables from the input data, including: exhaustive or heuristic search algorithms, genetic algorithms, evolutionary computation techniques, forward propagation and backward elimination strategies, or various hybrid strategies combining the above [94, 406]. The forward propagation (sometimes referred to as sequential forward selection or addition) strategy starts with the empty set and consecutively adds one attribute at a time until certain criteria are met [59]. On the other hand, the backward elimination strategy starts with the full set (or relatively large set of attributes that satisfies required criteria). In each iteration, one attribute is removed - as long as the reduced set satisfies given criteria. Those algorithms which aim to obtain the possibly minimal set of attributes usually combine the heuristic search or forward selection with the subsequent phase of backward elimination [183, 412]. We may also refer to a number of studies on parallelization of feature selection algorithms, e.g., by exploiting the computational capabilities of modern heterogeneous systems that contain several CPUs and GPUs [129], or by using Map Reduce paradigm and Spark framework [283]. Big Data aspects of attribute granulation and selection are discussed further in Section 4.2.3.

Depending on whether the training set is labeled or not, feature selection algorithms can be categorized into supervised, unsupervised, and semi-supervised. Given the input data as a table with m samples and n features $A = \{a_1, .., a_n\}$, and the target variable d, the supervised feature selection problem is to find a sub-set of features $R \subseteq A$ that "optimally" characterizes d [294]. Unsupervised feature

selection is a less constrained search problem (without class labels), often depending on clustering quality measures [431], statistical and information measures [430], or on various hierarchical and granular structures – as briefly discussed further in Section 2.4. A comprehensive review of unsupervised methods can be found in [12]. It is, however, quite common to have a data set with huge dimensionality but a small labeled sample size. Under the assumption that, both, labeled and unlabeled data are sampled from the same population generated by the target concept, the semi-supervised feature selection methods make use of both labeled and unlabeled data to estimate the relevance of evaluated features [341]. One way to do this is to transform the partially labeled data into completely labeled. Whereas, the other approach is to construct a measurement to cover both labeled and unlabeled data. For this purpose, one may use ensemble selectors, for example based on rough set based local neighborhood decision error rate [245], or may incorporate additional knowledge, like graph-based structures, into semi-supervised FS methods [342].

Feature selection methods can be further categorized into three main groups: wrapper, embedded, and filter methods [39,148]. Wrapper methods make a selection of attributes based on the results of a preliminary data analysis. Wrappers use the learning algorithm as a part of the feature subsets evaluation, i.e., classification (or regression) model is used as a black box for assessing the feature subsets usefulness in terms of the error (or fitness) rate obtained by a wrapped model on a testing set. Wrapper methods include simple approaches, like greedy sequential searches, but also more elaborate algorithms like recursive feature elimination, or evolutionary and swarm intelligence algorithms [169,254]. Although these techniques may lead to feature subsets well corresponding to the analyzed problem, they require training a model for a combinatorial number of times, hence the computational cost becomes prohibitive for high dimensional data sets. Embedded methods are nested in machine learning algorithms, and incorporate knowledge about the specific structure of the class of functions used by a certain learner, e.g., bounds on the leave-one-out error of SVMs [30]. Other examples are: Lasso regression, classification and regression trees, or gradient boosting [35]. Embedded methods are usually less computationally expensive, still are much slower than filters. Same as in the case of wrappers, the selected features are dependent on the learning machine.

In contrast to the above-discussed methods, filters carry out the attributes selection regardless of the chosen model, since for the assessment of feature subsets, they use evaluation metrics independent of the induction learning algorithm [229]. This strategy is particularly useful because of its efficiency. Typically, attributes are ranked according to various types of scores, and those with the highest scores are used to train the model, with an implicit assumption that choosing appropriate attributes improves the accuracy and efficiency of classification or regression. By applying statistical measures, one can find columns that do not contribute to the accuracy of a model (or might in fact decrease its accuracy) and remove them before the final training phase. Filter methods can be roughly classified further by the filtering measures they employ to heuristically determine the subset of attributes with the highest predictive power, i.e., information, distance, dependence, consistency, similarity, or statistical measures. Examples of which include univariate criteria like: correlation between evaluated features and a target variable [153,174], entropy,

chi-square, analysis of variance (ANOVA), or other statistical tests [39], as well as multivariate tests like various attribute dependency measures from the rough set theory (RST) [75,183,367], which are discussed in detail in Section 3.1.

Using the correlation coefficient is a simple yet effective approach to FS [153,174]. For the attribute $a \in A$ and the decision d, Pearson's correlation coefficient $r_{a,d}$ (or $r_{a,b}$ in the case of correlation between two attributes $a, b \in A$), for data with m samples, is defined as:

$$r_{a,d} = \frac{\sum_{i=1}^{m}(a_i - \bar{a})(d_i - \bar{d})}{\sqrt{\sum_{i=1}^{m}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{m}(d_i - \bar{d})^2}}$$

where $\bar{a}$ and $\bar{d}$ are the mean values for the investigated attribute and the decision, respectively. Whereas, a_i and d_i are the values of the attribute a and the decision d for the i-th sample.

Another simple yet very popular test to maximize the relevance of selected features is mutual information. Given two random variables X and Y, and their probabilistic density functions p(x), p(y), and p(x,y), mutual information is defined as:

$$I(X;Y) = \int_Y \int_X p(x,y) \log \frac{p(x,y)}{p(x), p(y)} dxdy$$

Given m samples, we may approximate density functions $\hat{p}$ for the attribute $a \in A$ as:

$$\hat{p}(a) = \frac{1}{|A|}\sum_{i=1}^{m}\delta(a - a_i, h)$$

where a_i is the value of the attribute a for the i-th sample, $\delta(.)$ is the density estimator (e.g., Parzen window function for which h is the window length) [101]. Naturally, we may consider mutual-information-based feature selection for both discrete and continuous data. For discrete (categorical) variables probability tables can be estimated from data samples with the following formula:

$$I(X;Y) = \sum_{y \in Y}\sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x), p(y)}$$

Mutual information may be equivalently expressed with entropy[8] as:

$$I(X;Y) = H(X) - H(X|Y)$$

where $H(X|Y)$ is the conditional entropy. Other well-known feature ranking strategies are Fisher Score that optimize between-class variance and the within-class variance, or Relief-based algorithms that order features based on the nearest neighbor distance [380].

Among filter-based feature selection methods, the most interesting from our viewpoint are multivariate algorithms [376]. Such approaches rely on inter-feature

[8]Entropy is one of the basic measures of information contained in data. For a discrete random variable X with possible values $\{x_1, .., x_m\}$ is defined as: $H(X) = -\sum_{i=1}^{m} p(x_i)log(p(x_i))$.

Algorithm 1: Hybrid FS combining mRMR filter with a wrapper model

Data: U – samples, A – set of features, d – target variable;
N – max number of features to be evaluated during forward propagation
$\mathbb{M}^{\Phi}$ – selected ML algorithm for wrapper model
$\Phi^{\mathbb{M}}$ – selected criterion to assess model quality
k – cross validation parameter
Result: $R \subseteq A$ – selected attribute subset

```
/* Initialization                                                    */
```
$R \leftarrow argmax_{a \in A}\Big(I(a;c)\Big)$
$i \leftarrow 1$
$R_{best} \leftarrow R$
$\Phi^{\mathbb{M}}_{best} \leftarrow$ evaluation of $\mathbb{M}^{\Phi}(R,U)$ with $\Phi^{\mathbb{M}}$ under k-fold cross validation
$\Phi^{\mathbb{M}}_{candidate} \leftarrow 0$
```
/* Forward propagation                                               */
```
while $\Big(i < N\Big)$ **do**
  $i++$
  `/* incremental mRMR criterion */`
  $a = argmax_{a_i \in A \setminus R}\Big(I(a_i;c) - \frac{1}{i}\sum_{a_j \in R} I(a_i;a_j)\Big)$
  $R \leftarrow R \cup \{a\}$
  `/* evaluation with wrapper model */`
  $\Phi^{\mathbb{M}}_{candidate} \leftarrow$ eval. of $\mathbb{M}^{\Phi}(R,U)$ with $\Phi^{\mathbb{M}}$ under k-fold cross validation
  if $\Big(\Phi^{\mathbb{M}}_{candidate} > \Phi^{\mathbb{M}}_{best}\Big)$ **then**
    $R_{best} \leftarrow R$
    $\Phi^{\mathbb{M}}_{candidate} \leftarrow \Phi^{\mathbb{M}}_{best}$
  end
end
return R_{best};

dependencies when selecting a feature subset. Most of multivariate filtering algorithms attempt to avoid including unnecessary features by measuring redundancy within the selected subset. Such methods iteratively select features that provide the most relevant information regarding dependent variable values (e.g., are highly correlated or have a high value of mutual information index) and, on the other hand, are less dependent on the already-selected features [243]. As a result, they produce quite compact feature sets – what is a big advantage in practice [356]. For example, in [174], authors search for features which have strong correlations with a target class, yet uncorrelated mutually. This way implementing the correlation-based multivariate FS method to identify the most prognostic genes to classify biological samples of binary and multi-class cancers.

In terms of mutual information, feature selection algorithms aim to find a feature set $R \subseteq A$, containing features that jointly have the largest dependency on the target

variable d. One of the most prominent examples of the multivariate FS based on mutual information is minimum redundancy maximum relevance algorithm (mRMR) [90, 294]. The main objective of mRMR is to find the subset of features $R \subseteq A$ that maximizes the following criterion:

$$\max_{R \subseteq A} \Phi(R, d) = \frac{1}{|R|} \sum_{a_i \in R} I(a_i; d) - \frac{1}{|R|^2} \sum_{a_i, a_j \in R} I(a_i; a_j)$$

The objective of mRMR algorithm is to maximizes relevance between selected features and the decision (the left factor of the above subtraction), and to minimize the redundancy among selected features (the right factor of the above subtraction) [90]. In practice, we may use an incremental search strategy to find the near-optimal solution as shown in line 11 in Algorithm 1. As proposed in [294], mRMR criterion may be combined with wrapper FS method. In each iteration of the forward propagation the wrapper model $\mathbb{M}^{\Phi}$ is evaluated with k-fold cross validation on the given data sample and so far selected features, i.e., $R^{(1)} \subset R^{(2)} \subset R^{(3)} \subset \cdots \subseteq A$ to assess the predictive quality of candidate feature set – as presented in Algorithm 1.

To provide more examples of multivariate methods, we may further refer to N-MRMCR-MI method based on the normalization of maximum relevance and minimum common redundancy for the nonlinear optimization problems [61]. There are more approaches that rely on mutual information, e.g., maximum relevance minimum multicollinearity (MRmMC) [334], double input symmetrical relevance filter (DISR), or normalized joint mutual information maximization (NJMIM) [31]. We may also recall a linear feature selection method called dynamic change of selected feature with the class (DCSF) that employs both mutual information and conditional mutual information [117], which eliminates irrelevant and redundant features by introducing the dynamic information change of already-selected features with the class. In [156], authors propose two FS algorithms and evaluation criterion inspired by mutual information, ReliefF, and Fisher score. Naturally, there are many more multivariate filters [29, 53], or combinations of filters and wrappers [123, 155]. In [39, 383, 391, 406], a comprehensive review of more related methods can be found.

Most of feature selection approaches are focused on achieving possibly compact data representation to perform efficiently on large data volumes [374], or to scale with respect to high dimensionality [243]. However, as in real life applications data may be processed continually over time, and some features may become temporarily unavailable or unreliable, it is also worth to study various extensions of standard feature selection algorithms, including such aspects as: incomplete data handling [76], dynamic and incremental data processing [192], or feature cost analysis [263]. To some extent, the ideas presented in this dissertation could be compared to the notion of stability (or robustness) of selected feature subsets [198, 282]. Stability of feature selection techniques can be expressed as a variation in feature selection results due to changes in the data, e.g., when training samples are added or removed [329]. If the FS algorithm produces a significantly different subsets for any perturbations in the training data, then that algorithm becomes unreliable. Measuring stability of selected features is particularly important in biological and medical research, indicating whether the selected features are likely to be a real clinical signals worth further investigation, or not [127]. There are two popular approaches to assess the stability

of particular FS algorithm: a similarity-based approach and the frequency-based approach. In both cases, we may measure the stability of a given feature selection algorithm as the variability of its output with respect to data sampling [281].

Let $\mathbb{R} = \{R_1, .., R_M\}$ be the set containing M feature subsets $R_i \subseteq A$ being results of M consecutive runs of the evaluated FS algorithm, e.g., on different data sub-samples. In the frequency-based approach, we interpret the feature selection results $\mathbb{R} = \{R_1, .., R_M\}$ as a binary embedding, where 1 means that feature a_i has been selected, whereas 0 means the opposite. For a given data containing $|A| = n$ features, we may represent $\mathbb{R}$ in a tabular form as:

$$\mathbb{R} = \begin{pmatrix} selected(a_1, 1) & \cdots & selected(a_n, 1) \\ selected(a_1, 2) & \cdots & selected(a_n, 2) \\ \vdots & \ddots & \vdots \\ selected(a_1, M) & \cdots & selected(a_n, M) \end{pmatrix}$$

where i-th row corresponds to subset $R_i \in \mathbb{R}$ selected in i-th algorithm run, and the $selected(.,.)$ function for an attribute $a \in A$ and i-th algorithm iteration is defined as:

$$selected(a, i) = \begin{cases} 1 & \text{if } a \text{ was selected in i-th FS algorithm run} \\ 0 & \text{opposite} \end{cases}$$

The observed frequency of selection of a feature $a \in A$ after M algorithm runs may be defined as:

$$\hat{p}(a) = \frac{1}{M} \sum_{i=1}^{M} selected(a, i)$$

Here, one can define the stability measure by the selection frequencies of each feature after M algorithm runs, e.g., as the frequency of selection averaged over all features [127]:

$$\hat{\Phi}(\mathbb{R}) = \frac{1}{|A|} \sum_{a \in A} \hat{p}(a)$$

Naturally, there are more frequency-based approaches to assess FS stability, for example: relative weighted consistency, or entropy of feature sets [281].

In the similarity-based approach, we define stability of algorithms as the average pairwise similarity between the possible pairs of feature sets in $\mathbb{R}$:

$$\hat{\Phi}(\mathbb{R}) = \frac{1}{|\mathbb{R}|(|\mathbb{R}| - 1)} \sum_{R_i \in \mathbb{R}} \sum_{\substack{R_j \in \mathbb{R} \\ R_j \neq R_i}} Sim^{\phi}(R_i, R_j)$$

where the stability measure $\hat{\Phi}$ depends on the similarity measure Sim^{ϕ} of a choice, which may be a Hamming ($Sim^{Hamming}$) or Dice-Sørensen (Sim^{Dice}) index, fuzzy similarity measures, e.g., generalized weighted Jaccard similarity (Sim^{GWJS}) [343], or RST based similarity measures, e.g., based on discernibility Sim^{Disc} (4.4) as presented later in Section 4.2.1, and many others [176]. For example, given the Jaccard index as

the similarity measure ($Sim^{Jaccard}$), the stability measure $\hat{\Phi}$, defined as the average pairwise similarity between the possible pairs of features, is as follows:

$$\hat{\Phi}(\mathbb{R}) = \frac{1}{|\mathbb{R}|(|\mathbb{R}|-1)} \sum_{R_i \in \mathbb{R}} \sum_{\substack{R_j \in \mathbb{R} \\ R_j \neq R_i}} \frac{|R_i \cap R_j|}{|R_i \cup R_j|}$$

Stability was investigated from various perspectives, e.g., by means of avoiding over-fitting [282], or minimizing an impact of data noise [14]. In the case of our study, the meaning of stability may refer to minimizing a risk of information insufficiency subject to a loss of access to some of pre-selected features.

Another thread of research that corresponds to robust / stable FS [224] is related to ensemble-based feature selection [45, 329]. An ensemble (sometimes referred to as a committee) is collection of single classification (or regression) models whose predictions are aggregated, e.g., by majority voting [217]. To address this aspect while building classifier ensembles, meta-learning algorithms, such as boosting or bagging, can be used. Ensembles for feature selection might be further classified following many diverse criteria [37], but the most simple division is into the homogeneous ones, in the case of which the base selectors are all of the same kind, and the heterogeneous ensembles that combine outputs of diverse FS methods. Diversity of ensamble-based FS may be investigated from many perspective, as thoroughly discussed further in Chapter 4. Yielding in an improved prediction performance, as confirmed in a detailed evaluation on real data sets presented further in Sections 5.1 and 5.2, is one of the main reasons to use an ensemble method with divers components. It would not make any sense to build an ensemble in which all the components offered the same result.

Ensembles have been shown to be an efficient way of improving predictive accuracy or/and decomposing a complex learning problem into easier sub-problems. The ensemble feature selection may be interpreted two-fold. On the one hand, several feature selection processes may be carried out (either using different training sets, different FS methods, or both), with the final goal to produce a single feature set as a combination of particular ensemble components [3]. In this approach, the aggregated predictions are expected to obtain more accurate and stable results, hence reducing the risk of choosing an unstable subsets, what is especially important for non-stationary environments [217], such as imbalanced data streams [49], or in the presence of concept drifts [85]. Indeed, merging multiple feature subsets obtained using ensemble techniques can yield results that are robust (or stable) from the above viewpoint. This kind of merging can actually lead toward establishing feature sets that induce high-quality prediction models [138]. However, robustness with regard to small data changes is not the same as robustness with regard to losing some of data dimensions. This latter aspect is specially relevant in knowledge discovery, and even more in those cases in which data dimensionality is very high, but the number of samples is not such, as they are more sensible to generalization problems [220].

On the other hand, we may apply FS for several times in order to produce the diversity of subsets for the purpose of subsequent ensemble learning methods. Here, ensemble-based methods in feature selection can be considered by means of creating multiple prediction models [89], whereby each model is built over a different subset of features, e.g., by constructing a rule-based classifier for each selected feature

subset and aggregating results of such predictors [361]. In [299], it was noted that ensemble construction based on random subspace selection can partially solve the missing feature problem, which is exactly what we want to address in our resilient feature selection framework presented in Chapter 3. Analogous approaches can be found also in the rough set literature, with respect to both, standard reducts [166] and approximate reducts [397]. Actually, using a variety of approximate reducts to construct an ensemble of diversified models can be efficient in many areas [356]. Still, when comparing to the approach to resilient feature selection, introduced further in Chapter 3, those aforementioned ideas – based either on merging multiple reducts [359], or on treating them as an ensemble [361] – are heuristic methods that miss explicit mathematical formulation of the properties of resilient feature sets and explicit optimization goals for algorithms aimed at searching for such sets in the data.

2.4 Information Granulation in Feature Extraction

Granular Computing (GrC) arose as a synthesis of insights into human-centred information processing that mimics human, intelligent synthesis of knowledge from information [23, 416]. Currently, information granulation plays an important role in modern machine learning and knowledge discovery algorithms, with a number of successful applications in various domains [355, 393, 426]. In this section, we focus on feature space granulation approaches introduced by now. Our objective is to provide a general overview of GrC, and to identify the main items on its agenda associating their usage in the setting of feature extraction. This way, we lay foundation for our approach by explaining how the granules can be formed, interpreted, and utilized by feature extraction algorithms.

Decision support in solving problems related to complex systems requires relevant computation models as well as methods for reasoning [348]. In recent years, one can observe a growing interest in the area of GrC as a methodology for modeling and conducting complex computations, in various domains of information technology, machine learning, and feature extraction, in particular [139]. On the other hand, human-centricity comes as an inherent feature of intelligent systems [293]. It is anticipated that a two-way human-machine communication is imperative, and interactive communication of intelligent systems with users becomes substantial [175].

The possibility to take advantage of additional domain knowledge provided by human experts relies on the observation that human thinking and perception in general, and their reasoning while performing data exploration tasks in particular, can comprise different levels of abstraction, display a natural ability to switch focus from one level to another, or operate on several levels simultaneously [279]. Human, however, perceives the world, reasons, and communicates at some level of abstraction that, unlike information systems and algorithms, comes hand in hand with non-numeric constructs. Those embrace collections of entities characterized by some notions of closeness, proximity, functionality, resemblance, or similarity, referred to as *information granules* (*granules* or *infogranules*, for short) [292, 413].

The construction of a granular system for a given data set is frequently portrayed

as a procedure of zooming in and out on the data or, in other words, changing the data "resolution". Depending on the chosen level of granularity, some data items (objects, cases, instances) become indistinguishable. Hence, the "length" of the data is altered, which corresponds to possible reduction of the storage and processing resources. By employing compact descriptions of granules – defined as collections of original data elements gathered together – one can accelerate computations and, moreover, make the results of those computations more meaningful for domain experts. It is also worth mentioning that the idea of zooming in and out – i.e., switching between different levels of information granularity – is popular in the area of analytical processing in databases [355]. However, one should remember that *data granularity* can have different meanings. In traditional databases, by *granular* data one usually means the most detailed, low level, exact data representation [11].

The granular approach to dealing with information systems does not have to be limited to just the length/volume dimension of the data set. It can also be used to modify, reduce and transform the "width" and "depth" of information. In GrC this is sometimes called *variable granulation* and *concept granulation*. Just like in a case of the "classical" granulation, where data objects are combined into more complex entities, attributes in data can be granulated by using similarity, distance or correlation between them. In particular, by constructing granules over the space of attributes in the data set it is possible to reduce dimensionality. In the simplest form it can be used to replace multiple features/dimensions by just the representative one of the corresponding granule. A more complex, yet still similar approach is represented by a reduction based on an information function and discernibility, typical for the theory of rough sets, where the original set of attributes is replaced by a reduct, i.e., a subset that carries the same amount of important information.

In the context of attribute granulation, two attributes are usually regarded as similar if they convey similar information about objects described in the data. For instance, one may consider similar two attributes whose values in the data are highly correlated. In fact, Pearson and Spearman correlation coefficients are commonly used as measures of attribute similarity for the purpose of attribute clustering [86]. There are, however, some other possibilities as well. For instance, further in this dissertation we examine an idea of building similarity of attributes by means of their ability to replace each other in the constructed decision models. Namely, if an attribute can be replaced by another without losing important information about investigated objects, it means that they complement in the same way the remaining attributes. The more generic approach is to search for whole feature sets with a guarantee of providing sufficient predictive power even if some of their elements are dropped [137, 142].

The proximity of attributes may have a few meanings as well. Typically, this term is used as a synonym of similarity. However, when it comes to granules of attributes, it may also be understood as a "physical proximity". For example, in coal mines, there are many sensors monitoring the safety of miners, which constantly gather data about the conditions underground [179]. When analyzing this type of data, it is important to consider locations of sensors, since readings from closely co-located devices are inherently correlated [15]. Moreover, events observed by one group of sensors are detected by other groups after some time and the delay, as well as the order, in which different sensors denote the event, often corresponds to the

ventilation scheme of the mine. For this reason, as noted in [277], it is often worth to consider the whole chunks of attributes corresponding to such proximate sensors. In this way, it is not only possible to improve readability of the resulting decision models, but also increase the performance of the whole data processing chain due to a more efficient utilization of local buffers for reading data streams [136,144]. Another practical consideration is the aspect of model robustness and fault tolerance. In this context, proximity of attributes may be regarded as a degree of dependency on a specific hardware equipment [179]. For instance, if one sensor is faulty, all attributes whose values are dependent on its readings will be unreliable [142].

It may also be desirable to consider granules of attributes that share some higher-level properties or that are tied by constraints imposed by a given application area [130]. Typically, domain experts associate such attributes with similar functionalities of investigated objects. Let us consider an example of the brain MRI data set investigated in [393], whereby features derived using some parameterized image processing procedures may be associated with groups of attributes that take different values for particular objects (these values depend on particular parameter settings) but describe the same aspect of the data. Another example of this type of situation is apparent in the analysis of a stock market. Many financial experts use technical indices to describe the behavior of stock prices in time. Such indices (e.g., moving averages, moving variance, RSI, TDI, stochastic oscillators, etc.) have many parameters, such as the considered time window size. Over long periods, the accuracy of time series model forecasting is invariably affected by interval length, and formulating effective interval partitioning methods can be very difficult. In [63], an interdisciplinary review of the idea of granularity in economic decision making from different perspectives, including: psychology, cognitive science, complex science, as well as behavioral and experimental economics is discussed.

The above considerations lead toward several observations. To begin, the spaces of features/attributes that require to be granulated can be more complex than a set of columns in a tabular data. The above considerations lead toward the observation that the spaces of features/attributes A [9] that require to be granulated can be more complex than a set of columns in a tabular data. In some real-life scenarios, the set A may require granulation because of its high cardinality. An example of such situation can be found, e.g., in [134], where an interactive GUI-based approach for grouping genes-attributes was introduced. However, in other scenarios the set A may not exist in a materialized form. We can rather think about a set A^* gathering all derivable attributes/features, e.g., wavelet coefficients in the case of EEG signal analysis [372] or JSON-driven aggregates defined for a semi-structured data set [186]. Thus, one could think about A^* as a space of all outcomes of the feature engineering/extraction techniques applied in a given application area. We shall treat A^* (sometimes taking a simple form of A) as our granulation domain.

The second observation is about the meaning of granules built over A^* (or A) from the perspective of data understanding and decision model construction, including feature selection. With respect to data understanding, it is implicitly assumed that features dropping into the same granules should be assessed by domain experts as having some kind of common background, by means of physical,

[9] Typically the set of all features/attributes is denoted with A [288]

functional or information-specific comparability. In particular, the information level of comparability may correspond to the way, in which particular features contribute to decision models aimed at classifying or distinguishing between different states of target variables. This aspect, as previously mentioned, seems to be close to the ideas of adapting various data clustering methods for the purpose of grouping together similarly acting or replaceable/interchangeable attributes [3, 177]. However, we also need to remember that all of the above flavors of similarity need to be coupled with some tangible criteria for assessing the quality of pre-defined or produced granules, especially in the context of feature extraction [185].

Identification of subgroups of similar variables is especially important for high-dimensional data exploration [40, 146]. In this context it is frequently useful to apply the modern algorithms aimed at big data clustering. Several instance clustering algorithms, like the expectation maximization or k-means, have already been implemented in the scalable environments [20, 78]. There are also some prior results reported on the feature clustering algorithms that are of particular interest in this paper. The hierarchies of granules/groups of features can be constructed using some interactive clustering methods as well [134]. It is also important to realize that the feature similarity measures employed in the above clustering approaches should somehow correspond to the ultimate goal of finding the groups of attributes that can play mutually comparable roles in the constructed decision models [3].

The demand for efficiency and effectiveness in Big Data scenarios resulted in a number of approaches to massively parallel feature reduction [307, 429], as well as highly scalable instance selection and deduplication [371, 377]. Popular code libraries like Spark or Mahout provide parallel implementations of well-known feature selection methods [104]. There are also approximate implementations of standard algorithms, which derive heuristic feature evaluation scores from granulated data summaries [60]. The speed of the feature and instance selection processes becomes especially important in interactive approaches [358], whereby, additionally, granular hierarchies of attributes may help the users to navigate through rich feature spaces. Introducing approximate computations into the feature selection processes is – in combination with making them highly parallel – an example of a more general trend in machine learning and knowledge discovery [7].

It is noteworthy that, just as for other popular feature selection methods, there were some interesting attempts to perform RST reduct derivation within the MapReduce framework [236]. The ideas of scalable performance of feature extraction, in particular reduct calculation [138], are most commonly related to decomposing computations with respect to rows/instances [163]. However, by introducing the elements of granulation into the feature spaces we can additionally scale up the algorithms in an "attribute-oriented" fashion. Surely, such granulation-related ideas could be considered – besides the algorithms originating from the theory of rough sets – within the scope of other popular feature selection/engineering solutions as well [148, 294].

Besides the so-far-mentioned rough sets [285, 288], there are numerous formal frameworks of information granules [293]. Let us recall some selected alternatives. Among the most popular ones we may point out the set theory, interval analysis [268] and fuzzy sets which deliver an important conceptual and algorithmic generalization

Table 2.2: A decision table $\mathbb{S}$ that is used in further illustrations in the frame of this study.

$U \setminus A$	a_1	a_2	a_3	a_4	a_5	a_6	d
u_1	*false*	'a'	$\bigtriangledown$	$\circ$	$\square$	'x'	*good*
u_2	*false*	'b'	$\bigtriangledown$	$\odot$	$\bullet$	'x'	*good*
u_3	*false*	'c'	$\bigtriangleup$	$\oslash$	$\diamond$	'x'	*good*
u_4	*false*	'd'	$\bigtriangledown$	$\otimes$	$\triangleleft$	'x'	*good*
u_5	*false*	'e'	$\bigtriangledown$	$\ominus$	$\star$	'y'	*good*
u_6	*false*	'f'	$\bigtriangledown$	$\oslash$	$\triangleright$	'y'	*good*
u_7	*true*	'g'	$\bigtriangleup$	$\otimes$	$\square$	'y'	*bad*
u_8	*true*	'h'	$\bigtriangleup$	$\ominus$	$\bullet$	'z'	*bad*
u_9	*true*	'i'	$\bigtriangledown$	$\oplus$	$\diamond$	'z'	*bad*

of sets by admitting partial membership of an element to a given information granule [415]. Shadowed sets distinguish among elements, which (i) fully belong to the concept, (ii) are excluded from it, and (iii) their belongingness is completely unknown [291]. The list of formal frameworks is quite extensive [287], interesting examples are also rough-fuzzy and fuzzy-rough sets [100], probabilistic sets [162], probabilistic rough sets [248], axiomatic fuzzy sets [246], or three-way decisions [411] under dynamic granulation [306], and many more [69, 427]. In this dissertation, however, we mainly focus on the interactive feature extraction methods related to the theory of rough sets [139, 142]. In the next sections, we discuss the advantages of pre-grouping of attributes from the perspective of feature selection, with the reduct-based decision models originating from the theory of rough sets [367].

2.5 Rough Sets Methods for Feature Selection

One of the data exploration methodologies where a large emphasis is put on the granulation of attribute space and multivariate feature selection is rough set theory (RST) [249,347]. RST as a whole provides a formalism for reasoning about imperfect data, handling such problems as data veracity, uncertainty, or incompleteness [154, 157, 219]. Its fundamental concept related to feature selection – and particularly dimensionality reduction – is a decision reduct, which is an irreducible subset of attributes (features, columns) that determines a target variable (so-called decision attribute) at the same level as the whole set of considered attributes.

In RST, we assume that the whole available information about an object $u \in U$ is represented in a structure called an information system [288] – a tuple (U, A), where U is a finite, non-empty set of objects, and A is a finite, non-empty set of attributes. Let us distinguish a decision attribute (class feature, target variable), which defines a partitioning of U into disjoint sets representing decision classes (or categories) that we want to describe using other attributes. An information system with specified decision attribute is called a decision table (or decision system) and is denoted by $\mathbb{S} = (U, A \cup \{d\})$, $A \cap \{d\} = \emptyset$.

For a given decision system $\mathbb{S} = (U, A \cup \{d\})$, one considers functions $a : U \to V_a$, $a \in A$, where V_a is the set of values of a. Such functions allow us to represent

$\mathbb{S}$ as a table with rows labeled by objects, columns labeled by attributes, and cells corresponding to pairs (u, a) assigned with values $a(u) \in V_a$ (see Table 2.2). Obviously, this kind of tabular representation is one of many equivalent formats of representing the data [398].

The indiscernibility relation (IND) expresses the fact that due to a lack of information (or knowledge) we are unable to discern some objects employing available information. In general, we are unable to deal with each particular object but we have to consider granules (clusters) of indiscernible objects as a fundamental basis for RST. Let us define indiscernibility relation $IND(R) : U \times U$, for any $R \subseteq A$, as follows [288, 304]:

$$IND(R) = \{(u_i, u_j) : \forall a \in R, a(u_i) = a(u_j)\} \tag{2.1}$$

after considering the decision:

$$IND(R) = \{(u_i, u_j) : \forall a \in R, a(u_i) = a(u_j) \wedge d(u_i) \neq d(u_j)\} \tag{2.2}$$

By analogy, we can define a discernibility (or more precise R-discernibility) relation $DIS(R)$, as:

$$DIS(R) = \{(u_i, u_j) : \exists a \in R, a(u_i) \neq a(u_j)\} \tag{2.3}$$

after considering the decision:

$$DIS(R) = \{(u_i, u_j) : \exists a \in R, a(u_i) \neq a(u_j) \wedge d(u_i) \neq d(u_j)\} \tag{2.4}$$

(In)discernibility relations enable us to express dependencies among attributes at a more universal level. We may notice that indiscernibility and disceribility are equivalence relations. We denote an equivalence class of each object $u \in U$ as $[u]_A$.

An excessive amount of attributes in A provides a great potential for data-driven reasoning. However, many of those attributes may be dispensable, or could be irrelevant from the point of view of a given problem corresponding to d. In such situations, A-based information about objects in U needs to be simplified. Selecting informative sets of attributes is conducted by referring to the notion of a reduct [288].

Definition 1 (Reduct)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$ be a decision table. Subset $R \subseteq A$ is called a superreduct, if and only if it determines d within U, denoted as $R \Rrightarrow d$. Superreduct R is called a reduct, if and only if there is no proper subset $R' \subsetneq R$, which holds the superreduct condition.

From a formal point of view, we should write $\Rrightarrow_{\mathbb{S}}$ instead of $\Rrightarrow$, as the requirement of determining d by R is data-specific. However, we use a simplified notation whenever it does not lead to misunderstandings. Analogously, one may think about the usage of various heuristic measures while evaluating (subsets of) attributes in filter-based feature selection algorithms. There are plenty of interpretations of the reduct definition (1) that correspond to several other concepts and theorems like: (in)discernibility relations, the (in)discernibility matrix, or the positive region. Below we provide a short review of several significant and representative reduct interpretations.

The first example refers to the indiscernibility relation (eq. 2.2), which enables us to express dependencies among attributes.

Definition 2 (Reduct - by IND relation)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$ be a decision table. Subset $R \subseteq A$ is called a superreduct, if and only if $IND(R) \subseteq IND(A)$. Superreduct R is called a reduct, if and only if there is no proper subset $R' \subsetneq R$, which holds the superreduct condition.

Equivalently, we may say that $R \subseteq A$ is a decision reduct, if and only if it is an irreducible subset of attributes such that each pair of objects $u_i, u_j \in U$ satisfying the inequality $d(u_i) \neq d(u_j)$ is discerned by R.

Another reduct definition is related to the indiscernibility relation and its quotient set (i.e., is constructed by the equivalence classes of IND). A subset of features $R \subseteq A$ is called a decision superreduct iff for any object $u \in U$ the indiscernibility class of u relative to A is a subset of some decision class, its indiscernibility class relative to R should also be a subset of that decision class.

Definition 3 (Reduct - by equivalence classes of IND)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$ be a decision table. Subset $R \subseteq A$ is called a superreduct, if and only if $[u]_A \subseteq [u]_d \Rightarrow [u]_R \subseteq [u]_d$. Superreduct R is called a reduct, if and only if there is no proper subset $R' \subsetneq R$, which holds the superreduct condition.

The next example refers to the discernibility relation. The numeric $Disc$ measure is based on the arity of discernibility relation:

$$Disc(R) = |\{(u, u') : \exists a \in R, a(u) \neq a(u') \wedge d(u) \neq d(u')\}| \tag{2.5}$$

The definition of the decision reduct by $Disc$ measure would be very similar to the above listed. The only difference would refer to the superreduct condition, which, for $Disc$ measure, would be defined with the following equation:

$$Disc(R) = Disc(A) \tag{2.6}$$

Another popular reducts formulation refers to the notion of function $\gamma : \mathcal{P}(A) \rightarrow [0, 1]$, which is commonly used to express a degree of dependence between a subset of attributes and the decision:

$$\gamma(R) = \frac{|POS(R)|}{|U|} \tag{2.7}$$

where POS denotes the positive region induced by R [288]:

$$POS(R) = \{u \in U : \forall_{u' \in U} d(u) \neq d(u') \Rightarrow \exists_{a \in R} a(u) \neq a(u')\} \tag{2.8}$$

For a decision system $\mathbb{S} = (U, A \cup \{d\})$, where cardinality of A and U is: $|A| = m$, $|U| = n$ we can define a discernibility matrix $\mathbb{M}(R)$. Discernibility matrices are useful for deriving possibly small subsets of attributes, still keeping the knowledge encoded within a decision system [349]. Each cell $c_{i,j}$ of $\mathbb{M}(R)$ for $i, j = 1..n$, $1 \leq i < j \leq n$ contains a list of attributes in $R \subseteq A$, which are discerning objects u_i, u_j with different decisions, or more formally:

$$c_{i,j} = \{a \in R \subseteq A : u_i, u_j \in U, u_i \neq u_j, a(u_i) \neq a(u_j) \wedge d(u_i) \neq d(u_j)\} \tag{2.9}$$

Among the presented variety of extensions of decision reducts, let us also discuss their approximate interpretations. Criteria for calculating approximate decision reducts are usually based on functions evaluating degrees of decision information induced by attribute subsets and thresholds for values of those functions' specifying which of those subsets are good enough. Such an approach may lead us to obtain subsets of attributes that are less accurate than exact reducts but could be preferred in some real-life applications to deal with large or noisy data, ultimately leading to smaller data representations.

For example, we may refer to α-approximations of reducts, where $\alpha \in (0, 1]$ is a real parameter [276]. The set of attributes $R \subseteq A$ is called α-reduct iff it is minimal in sense of set-inclusion, intersecting at least $\alpha \cdot 100\%$ of pairs of objects that are necessary to be discerned with respect to decision. More formally, we may define α-reduct with the following equation:

$$\frac{|\{c_{i,j} : R \cap c_{i,j} \neq \emptyset\}|}{|\{(u_i, u_j) : d(u_i) \neq d(u_j)|} \geq \alpha \tag{2.10}$$

We may also easily introduce the approximation threshold ε for many reduct criteria. For example, let us introduce it into criteria based on $Disc$ measure (2.6):

$$Disc(R) \geq (1 - \varepsilon) * Disc(A) \tag{2.11}$$

As a yet another significant example, we may point out approximate entropy reducts [352], in the case of which, the superreduct criterion relay on the conditional entropy $H(d|R) = H(R \cup \{d\}) - H(R)$. In the following specification, H plays the role of a penalty measure, which, with the given approximation threshold ε, corresponds to (H, ε)-approximate reducts introduced in [352].

Definition 4 (Reduct - by conditional entropy)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$ be a decision table. Subset $R \subseteq A \subseteq A$ is called a (H, ε)-approximate superreduct, if and only if $H(d|R) \leq H(d|A) - \log_2(1 - \varepsilon)$. Superreduct R is called a (H, ε)-approximate reduct, if and only if there is no proper subset $R' \subsetneq R$, which holds the superreduct condition.

There are a lot more extensions for approximate and exact reducts, often hybridized with other methods [304, 367]. For instance, in [183] a combination of iterative filter-based feature selection with statistical significance tests based on random probes and a typical RST-based redundant feature elimination was applied to calculate dynamically adjusted approximate reducts (DAAR). To provide a wider range of reasoning strategies, we could easily refer to the *rough membership function* $\mu_{d/R} : V_d \times V_R^U \to [0, 1]$ [286], *majority decision function* $m_{d/B} : V_R^U \to 2^{V_d}$ [351], and many others [53, 253, 340]. A similar mixture of iterative feature selection and reduction was considered in [80]. Although in that latter case authors did not refer to the rough set literature, their feature reduction phase actually follows the same criteria as RST-based reduct calculation methods referring to the discernibility of (almost all) pairs of objects having different target variable values.

Figure 2.2 presents the attribute lattice for the data in Table 2.2. In this context, reduct computation means the search through the lattice. A minimal reduct may be interpreted as the first subset for every path from $\emptyset$ to A that satisfies the

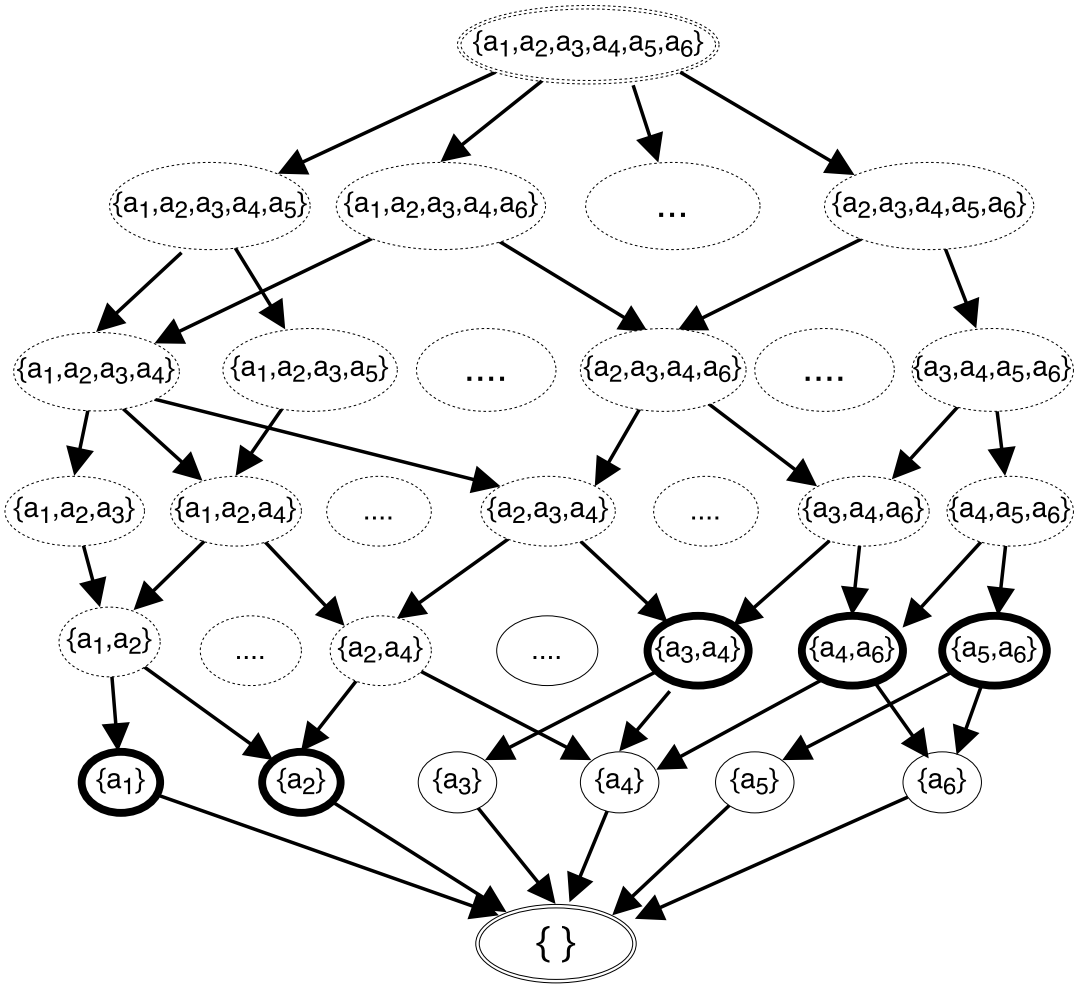


Figure 2.2: The lattice for the data in Table 2.2. Each oval corresponds to a single attribute subset, starting from the empty set at the bottom, ending with the whole set of attributes at the top. Bold ovals correspond to reducts, dotted ovals correspond to superreducts, arrows correspond to the set inclusion relation $\subset$.

considered criterion. We may refer to a number of well-established reduct search algorithms with this respect [118,319,412]. It is also important to stress out that the problems of finding various types of minimal reducts are known to be NP-hard and the RST-related literature reports a number of studies with this respect [92,349,360].

Chapter 3

Resilient Feature Selection

There is a variety of approaches for automatic feature selection [59]. Still, it is hard to find a method that would put together different aspects of feature subset quality, such as expected efficiency of the corresponding model, its interpretability from the viewpoint of the end-users, a risk of loss or lack of sufficient data to make decisions during long-term operations, and so on. In this Chapter, we concentrate on the last of the above-listed quality aspects. Our goal is to formulate new constraints, whereby selected feature sets are guaranteed to provide enough information about the considered target variables even if some of those features are temporarily dropped.

We formalize such constraints by introducing r-$\mathbb{C}$-reducts – irreducible subsets of features providing a satisfactory level of information about the target variable according to a given criterion function $\mathbb{C}$, even after removing r elements. The proposed approach is based on generalization of the notion of an approximate reduct known from the rough set theory (RST) [367]. This way, we continue RST-based research on resilient feature selection that was started in [137] by extending standard reducts [288]. However, the framework proposed in this paper embraces a much wider family of criteria specifying that a given feature subset is good enough to determine target variable values. We are actually able to refer to the whole realm of filter-based feature selection strategies [79], now defining a satisfactory feature set as the one whose evaluation function exceeds a certain threshold even after removing its r elements, $r \geq 0$.

In the feature selection process based on r-$\mathbb{C}$-reducts, an analyst should be able to control the level of resilience occurring in generated subsets of features while maintaining their relevance to the analyzed problem. In other words, the idea is to let an analyst achieve a relatively compact representation of the data tuned for the investigated problem, whereby the selected feature set should preserve its relevance even in case of partial data loss. However, to make this approach feasible, we need to investigate computational complexity of the corresponding search tasks. Then, we should also design efficient algorithms deriving meaningful r-$\mathbb{C}$-reducts from the data. The rough set literature is a good source of inspiration for both above aspects.

In RST, there is a lot of attention paid to NP-hardness of finding various versions of reducts in the data [270]. For example, in [351] it was proposed to evaluate feature subsets using a measure modeling accuracy of rule-based classifiers induced by those subsets over the training data. Then, the problem of finding minimal – in

terms of cardinality – subset of features providing ε-almost the same value of such measure as the whole set of features was shown to be NP-hard for an arbitrary fixed threshold $\varepsilon \in [0, 1)$. Problems related to r-$\mathbb{C}$-reducts are analogous. As the important theoretical contribution in this Chapter, we show that any NP-hard problem of finding minimal attribute subset that yields satisfactory level of information according to a given $\mathbb{C}$ remains NP-hard for an arbitrary resilience level r. As a special case, the task of finding minimal subset of features providing ε-almost the same level of the aforementioned accuracy measure as the whole set even after removing arbitrary r elements is NP-hard.

The second contribution is a broad study on algorithmic aspects of searching for minimal r-$\mathbb{C}$-reducts. By following a popular idea of dynamic exploration of the lattice of feature subsets, whereby some of its elements turn out to be labeled as satisfying the criteria for providing enough information while others do not, we elaborate on two generic strategies, namely, breadth first search (BFS) and depth first search (DFS). For BFS, we adapt the well-known Apriori algorithm [331] for the purpose of r-$\mathbb{C}$-reduct search. For DFS, we extend standard reduct construction methods [353] to incorporate resilience of generated feature sets. Our study includes also some illustrative examples of data sets, as well as the analysis of computational cost of particular algorithms.

The rest of the Chapter is organized as follows. In Section 3.1, we introduce the notion of criterion function $\mathbb{C}$, which enables us to consider various feature selection formulations at a higher level of abstraction with $\mathbb{C}$-reducts. In Section 3.2, we discuss the idea of resilient feature selection and, accordingly, we introduce r-$\mathbb{C}$-reducts. In Section 3.3, we outline an Apriori-inspired algorithm that generates all r-$\mathbb{C}$-reducts of a given type. In Section 3.4, we study the tasks of resilient feature selection from the perspective of their computational complexity. We prove that many NP-hard feature selection / elimination problems remain NP-hard for any arbitrary resilience level r. In Section 3.5, we present heuristic DFS algorithms for searching for optimal r-$\mathbb{C}$-reducts, with specific examples of permutation-based and approximation methods.

3.1 $\mathbb{C}$-reducts

In this section, we take a step towards a generalization of feature selection methods as a process of achieving a feature subset that satisfies expected criteria. In many cases, especially in data analysis, it is much more interesting whether the given feature subset complies with respect to the defined function that is verifying some specified criteria rather than the exact value of a quality (or error) measure. Below, we generalize this way of reasoning about attribute subsets by introducing criterion functions, which, for each given decision table $\mathbb{S} = (U, A \cup \{d\})$, return a binary assessment of the candidate attribute subsets.

Definition 5 (Criterion Function)**.**
A criterion function $\mathbb{C}$ is a function, which assigns, for any $\mathbb{S} = (U, A \cup \{d\})$, values 0 and 1 to the subsets of A (i.e., $\mathbb{C} : \mathcal{P}(A) \to \{0, 1\}$, where $\mathcal{P}(A)$ denotes the set of all subsets of A) in such a way that, for any $X, Y \subseteq A$, if $X \Rrightarrow Y$ then $\mathbb{C}(X) \geq \mathbb{C}(Y)$.

We write $\mathbb{C} : \mathcal{P}(A) \to \{0, 1\}$ instead of $\mathbb{C}_{\mathbb{S}} : \mathcal{P}(A) \to \{0, 1\}$. However, we go back to explicit data-specific notation in Section 3.4. Having this in mind, let us note that for any $X, Y \subseteq A$, if $X \supseteq Y$ then $X \Rrightarrow Y$, hence $\mathbb{C}(X) \geq \mathbb{C}(Y)$. Such monotonicity of $\mathbb{C}$ is illustrated in the attribute subset lattice in Figure 2.2. The above definition allows us to consider a very broad range of criteria, not all of which could be anyhow reasonable for feature selection. Still, once we conclude the particular approach does have a sense and is compliant to the presented generic definition (as we see in the following sections, there is a number of so far developed approaches that do comply), we may easily formulate its resilient versions and appraise their complexity.

Besides the constraint expressed for the selected R by $\mathbb{C}$, the proposed approach follows the very common for RST (but not only for RST – see, e.g., [80]) objective to achieve the smallest feature subsets – reducts. Below we extend the notion of a reduct with respect to $\mathbb{C}$.

Definition 6 (Criterion Reduct)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$ and $\mathbb{C}$ be given. Subset $R \subseteq A$ is called a $\mathbb{C}$-superreduct, if and only if $\mathbb{C}(R) = 1$. We call R a $\mathbb{C}$-reduct, if and only if, additionally, there is no proper subset $R' \subsetneq R$ such that $\mathbb{C}(R') = 1$.

In relation to the notions introduced in Definition 1, we may notice that they can be easily rephrased using specific criterion function, namely:

$$\mathbb{C}^{\Rrightarrow}(R) = \begin{cases} 1 & \text{if } R \Rrightarrow d \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Indeed, reducts and $\mathbb{C}^{\Rrightarrow}$-reducts are equivalent to each other. It is also worth adding that there are decision tables $\mathbb{S} = (U, A \cup \{d\})$ for which $\mathbb{C}^{\Rrightarrow}(A) = 0$. In RST, they are called inconsistent. In such cases, there are no (super)reducts in terms of Definition 1. Definition 6 is surely far more general than Definition 1, subject to a choice of $\mathbb{C}$. In the literature, one can find many variants of reduct definitions. Below, we recall some of the popular extensions reviewed in Section 2.5, and re-formulate them using their corresponding criterion functions.

The criterion function that encapsulates the reduct definition based on *Disc* measure (eq. 2.6) may be constructed as follows:

$$\mathbb{C}^{Disc}(R) = \begin{cases} 1 & \text{if } Disc(R) = Disc(A) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

The criterion function that encapsulates the reduct definition based on discernibility matrix $\mathbb{M}$ may be constructed as follows:

$$\mathbb{C}^{\mathbb{M}}(R) = \begin{cases} 1 & \forall_{1 \leq i < j \leq n}, c_{i,j} \in \mathbb{M}(A), c'_{i,j} \in \mathbb{M}(R), \text{ if } |c_{i,j}| > 0 \Rightarrow |c'_{i,j}| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

The criterion function defining so-called γ-reducts is as follows:

$$\mathbb{C}^{\gamma}(R) = \begin{cases} 1 & \text{if } \gamma(R) = \gamma(A) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The above examples can be generalized using the notion of a quality measure $Q : \mathcal{P}(A) \to \Theta$, where $(\Theta, \succeq)$ refers to a partially ordered set in which every two elements have a unique lower and upper bound [142, 275, 353]:

Definition 7 (Quality Measure)**.**
Function Q is called a quality measure if, for any $\mathbb{S} = (U, A \cup \{d\})$, it assigns the subsets of A with the elements of Θ (i.e., $Q : \mathcal{P}(A) \to \Theta$) in such a way that for any $X, Y \subseteq A$, if $X \Rrightarrow Y$ then $Q(X) \succeq Q(Y)$.

The above property of Q will be further referred to as the monotonicity with respect to functional dependencies, which yields in particular – like in the case of Definition 5 – the monotonicity with respect to set inclusion (cf. the lattice in Figure 3.2). In practice, the most commonly used specification of Θ are $\mathbb{R}$, $\mathbb{N}$, or $(0, 1]$ with $\geq$ relation. Similarly, as in the case of the criterion function definition (5), Definition 7 is intended to cover essential properties of feature subset measures to generalize the further discussion, not to implement a feature selection by itself.

The following criterion functions correspond to a number of Q-based definitions of so-called approximate reducts. A general mechanism is to use measures $Q : \mathcal{P}(A) \to [0, +\infty)$ together with an approximation threshold $\varepsilon \in [0, 1)$, which is responsible for the allowed degree of losing information while removing attributes from A:

$$\mathbb{C}^{(Q,\varepsilon)}(R) = \begin{cases} 1 & Q(R) \geq (1-\varepsilon) * Q(A) \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

Proposition 3.1.1. *For every Q satisfying conditions of Definition 7, for every $\varepsilon \in [0, 1)$, criterion function $\mathbb{C}^{(Q,\varepsilon)}$ satisfies conditions of Definition 5.*

Proof. Straightforward. □

For example, let us note that RST-based functions $Disc$ and γ can be considered as special cases of the above framework [275, 367]. The corresponding criterion function for $Disc$ measure would be defined as follows:

$$\mathbb{C}^{(Disc,\varepsilon)}(R) = \begin{cases} 1 & Disc(R) \geq (1-\varepsilon) * Disc(A) \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

The criterion function defining γ-reducts with the approximation threshold is as follows:

$$\mathbb{C}^{(\gamma,\varepsilon)}(R) = \begin{cases} 1 & \gamma(R) \geq (1-\varepsilon) * \gamma(A) \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

Proposition 3.1.2. *Functions Disc and γ satisfy conditions of Definition 7.*

Proof. It is known that both considered functions are monotonic with respect to set inclusion. The property related to functional dependencies can be shown analogously. □

Proposition 3.1.3. *Function $\mathbb{C}^{\mathbb{M}}(R)$ 3.3 satisfies conditions of Definition 7.*

Proof. Straightforward. Conditions of discernibility matrix are constructed basing on discernibility relation DIS and has similar behaviour as *Disc*. Compare Proposition 3.1.2. □

The above simple facts will be important while considering examples of $\mathbb{C}^{(Disc,\varepsilon)}$-reducts and $\mathbb{C}^{(\gamma,\varepsilon)}$-reducts. Let us refer to [188, 288, 351] for more examples of quality measures that can be utilized to specify $\mathbb{C}^{(Q,\varepsilon)}$-reducts.

Yet another option to utilize Definition 6 to express various variations of reducts is to consider functions modeling a lack of information about the decision attribute, such as, e.g., conditional entropy $H(d|R) = H(R \cup \{d\}) - H(R)$. In the following specification, H plays the role of a penalty measure, which, with the given approximation threshold ε, corresponds to (H, ε)-approximate reducts introduced in [352].

$$\mathbb{C}^{(H,\varepsilon)}(R) = \begin{cases} 1 & H(d|R) \leq H(d|A) - \log_2(1-\varepsilon) \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Surely, as in the case of $\Rrightarrow$ and $\mathbb{C}$, functions H, *Disc*, γ, etc., could be marked with additional subscript corresponding to specific $\mathbb{S}$, though we omit it for simplicity.

Proposition 3.1.4. *Criterion function $\mathbb{C}^{(H,\varepsilon)}$ satisfies conditions of Definition 5.*

Proof. Straightforward, like in the case of Proposition 3.1.2. □

Let us finish this section with several remarks on computational aspects of deriving reducts from the data. In some cases, when $\mathbb{C}(\emptyset) = \mathbb{C}(A)$, reduct computation is trivial. This can happen if there is either no subset of attributes satisfying the given criterion ($\mathbb{C}(A) = 0$), or every subset does it ($\mathbb{C}(\emptyset) = 1$). Nevertheless, the problems of finding various types of minimal reducts are known to be NP-hard. For instance, let us consider the problem of finding minimal already-mentioned α-reducts [276], which are actually equivalent to $\mathbb{C}^{(Disc,\varepsilon)}$-reducts for $\alpha = 1 - \varepsilon$ (see equation (3.5)). As another example, let us mention NP-hardness of the problem of finding minimal (H, ε)-approximate reducts (or $\mathbb{C}^{(H,\varepsilon)}$-reducts using the terminology of equation (3.8)) proved in [352]. For further formulations of NP-hard problems related to the search of ε/α-related approximate reducts let us refer to [270, 367]. We may also refer to a number of well-established search algorithms with this respect [183, 307, 319, 412]. Accordingly, those well-known methods of finding reducts can be reconsidered for the purposes of ℂ-reducts as well. This interpretation can be also compared to other search strategies applied in the area of feature selection [80].

3.2 r-ℂ-reducts

In applications such as threat detection or recommendation systems, classification models often have to work on incomplete data. Most of the studies on feature extraction focus on deriving useful and understandable parameters (variables, attributes, features) in order to achieve possibly simplest, yet accurate models [194]. However, almost none of the available methods takes into account that the data may be lost or temporarily unavailable for the analysis. In this regard, let us recall

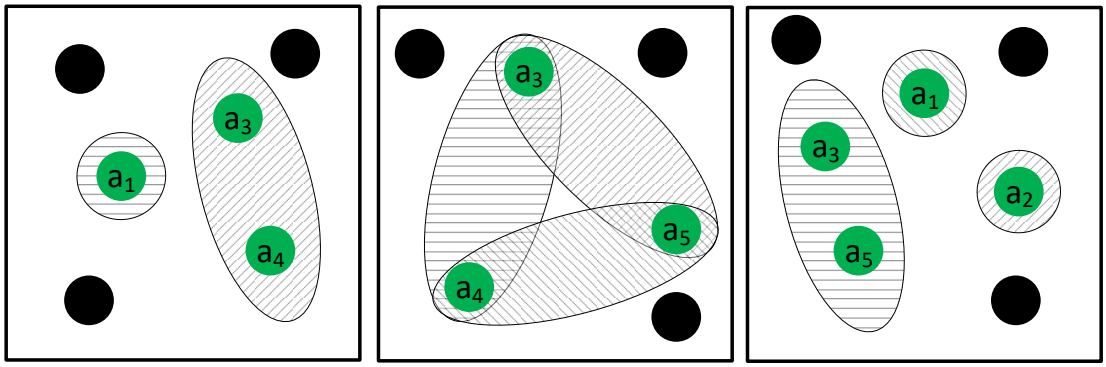


Figure 3.1: Three examples of r-reducts. Points represent attributes from Table 2.2, ovals are grouping attributes into standard reducts, the set of all labelled attributes included in reducts on each figure forms an r-reduct. The leftmost and the middle present examples of 1-reducts, since after removal of any attribute, the remaining attributes still form a superreduct. Analgously, the rightmost figure presents 2-reducts.

r-reducts [137] – one of the approaches to resilient feature selection that extends the concept of reduct to enable the governance of the redundancy level and, hence, to improve the resilience of the analysis. In Figure 3.1, a graphical interpretation of r-reducts is shown.

Definition 8 (r-reduct)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$ be a decision table. Subset $\breve{R} \subseteq A$ is called an r-superreduct, if and only if, after removing any $1 \leq n \leq r$ attributes $a_1, \ldots, a_n$ from $\breve{R}$, the remaining $R = \breve{R} \setminus \{a_1, \ldots, a_n\}$ is a superreduct. We say that $\breve{R}$ is an r-reduct, if and only if it is an r-superreduct and there is no proper subset $\breve{R}' \subsetneq \breve{R}$, which is an r-superreduct.

To emphasize the meaning of resilience let us consider the following scenario. Let us assume, for simplicity, that for each attribute in $R \subseteq A$, the probability that it is missing in the data during application of a prediction model is independent and equal to $p \in (0, \frac{1}{q * |A|})$, where $q > 1$ (q may refer to, e.g., the quality or price of utilized sensors). Then, for a standard version of (approximate) reducts the risk that the expected quality measure will not be satisfied is equal to p, while for r-reducts it is p^{r+1}.

To give a better understanding of r-reducts let us present all the subsets of attributes from the decision table in Table 2.2 as a lattice – starting with the empty set $\emptyset$ and ending with the full attribute set $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ – see Figure 3.2. We may imagine that some subsets of A have special properties that are retained by supersets – an example of such a property is discernibility of objects in a decision table $\mathbb{S} = (U, A \cup \{d\})$. Subsets: $\{a_1\}, \{a_2\}$, $\{a_3, a_4\}$, $\{a_4, a_6\}$, $\{a_5, a_6\}$ in Figure 3.2 correspond to reducts (compare Table 2.2) and the line marked as R0 corresponds to the border above which attribute subsets discern all the objects in the decision table.

Subsets $\{a_1, a_2\}$, $\{a_2, a_3, a_4\}$, $\{a_3, a_4, a_6\}$, $\{a_4, a_5, a_6\}$ presented in Figure 3.2 are 1-reducts. We may notice, that removal of any attribute from those sets guaranty discernibility of all objects, however if we remove two attributes it is not guaranteed – e.g., removal of $\{a_4, a_5\}$ from $\{a_4, a_5, a_6\}$. Thus, the line marked as R1 corresponds

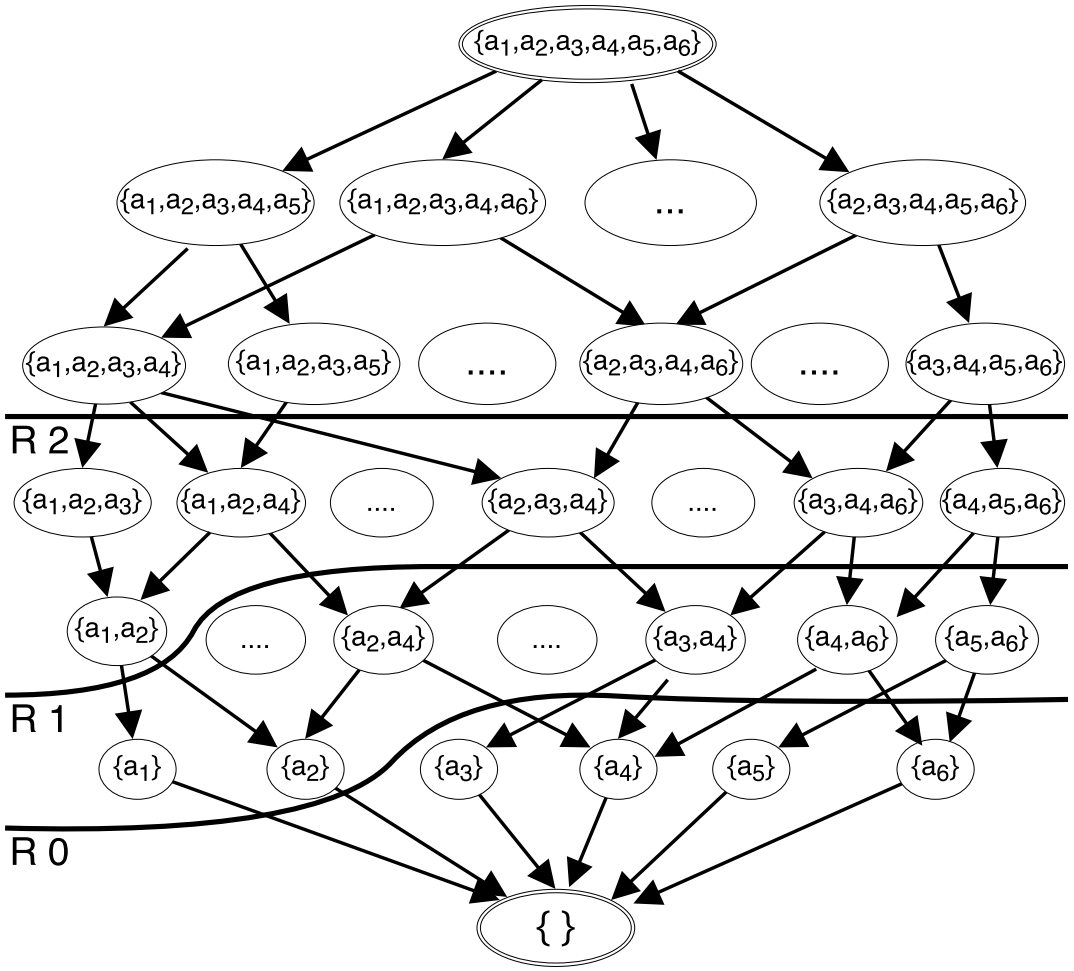


Figure 3.2: The lattice for the data in Table 2.2 with information about various resilience levels guaranteed by r-reducts. The line marked as R0 visualizes the border above which attribute subsets discern all objects in Table 2.2 with respect to d. The R1 line corresponds to the border above which each subset is a 1-ℂ-superreduct. That is, after the removal of any attribute the remaining attributes hold enough information to discern all the objects in Table 2.2. The R2 line visualizes subsets that are 2-ℂ-superreducts.

to the border above which set guarantee that after the removal any attribute the remaining ones discern all the objects. Similarly, the line marked as R2 corresponds to 2-reducts.

Using the criterion functions we may express the notion of, both, r-reducts and approximate r-reducts. Having defined the criterion function $\mathbb{C}$ for approximate reducts, e.g., $\mathbb{C}^{(Q,\varepsilon)}$ in equation (3.5), we may define a resilient criterion function r-$\mathbb{C}^{(Q,\varepsilon)}$ as shown in equation (3.9). Approximate r-reducts may be defined exactly the same way as presented in Definition 9. Therefore, in order to provide background for further discussions, let us reformulate r-reducts with the notion of criterion function. First, let us define the resilient version of criterion function r-$\mathbb{C} : \mathcal{P}(A) \to \{0, 1\}$ as:

$$r\text{-}\mathbb{C}(R) = \begin{cases} 1 & \text{if } \min_{R' \subseteq R : |R'| \geq max(|R|-r,0)} \mathbb{C}(R') = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Given the above, the resilient criterion reduct (r-ℂ-reduct) formulation is

straightforward:

Definition 9 (r-$\mathbb{C}$-reduct)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$, $\mathbb{C}$ and the expected resilience level r be given. A subset of attributes $\breve{R}$ is called an r-$\mathbb{C}$-superreduct, if and only if r-$\mathbb{C}(\breve{R}) = 1$. We say that $\breve{R}$ is an r-$\mathbb{C}$-reduct, if and only if it is an r-$\mathbb{C}$-superreduct and there is no proper subset $\breve{R}' \subsetneq \breve{R}$, which is an r-$\mathbb{C}$-superreduct.

Below we elaborate on some interesting properties of r-$\mathbb{C}$-reducts. Before doing this, let us just mention that in some special cases there is no risk of losing information, e.g., when $R = \emptyset$ is a reduct. Then we assume that $\breve{R} = \emptyset$ is an r-reduct for any r.

Proposition 3.2.1. *For every non-empty r-$\mathbb{C}$-reduct $\breve{R}$ there exist at least $r + 1$ reducts R for which $\breve{R} \cap R = R \wedge \breve{R} \cup R = \breve{R}$ (which means that r-$\mathbb{C}$-reduct may be expressed as a union of at least $r + 1$ $\mathbb{C}$-reducts).*

Proof. Let $\breve{R}$ be an r-$\mathbb{C}$-reduct in $\mathbb{S} = (U, A \cup \{d\})$. We put $\mathcal{R} = \{R \subseteq \breve{R} |\ R \text{ is a } \mathbb{C}\text{-reduct}\}$. Let $|\mathcal{R}| = k$ and $k \leq r$. Consider a set $X = \{a_1, \ldots, a_k\}$, such that for each $\mathbb{C}$-reduct $R_i \in \mathcal{R}$ there is $X \cap R_i \neq \emptyset$. Let us remove attributes $a_1, \ldots, a_k$ from $\breve{R}$. For the remaining set $R' = \breve{R} \backslash X$, for each $\mathbb{C}$-reduct $R_i \in \mathcal{R}$, there is $R' \not\supseteq R_i$. So, for every $R \subseteq R'$, R is not a $\mathbb{C}$-reduct. Hence, $R' = \breve{R} \setminus \{a_1, \ldots, a_k\}$ is not a $\mathbb{C}$-superreduct, whereas should be because $k \leq r$ and $\breve{R}$ is an r-$\mathbb{C}$-reduct. Contradiction. □

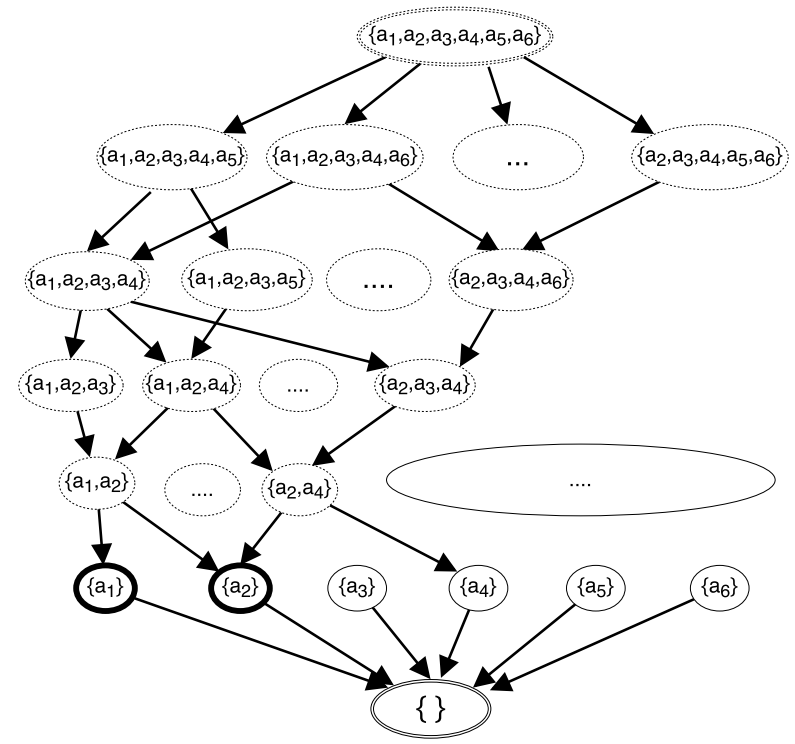
Proposition 3.2.2. *If in a given decision table $\mathbb{S}$ there exists a non-empty r-$\mathbb{C}$-reduct $\breve{R}$, for $r > 0$, then for each $a \in \breve{R}$ there exists $\mathbb{C}$-reduct R such that $a \in R$ and $R \subsetneq \breve{R}$.*

Proof. Let $\breve{R}$ be an r-$\mathbb{C}$-reduct and $\breve{R} = \breve{R}' \cup \{b\}$, where b is such that for each $R \subseteq \breve{R}$, if R is a $\mathbb{C}$-reduct, then $b \notin R$. For any r attributes $a_1, \ldots, a_r$ that satisfy $\{a_1, \ldots, a_r\} \cap \{b\} = \emptyset$, if we remove $\{a_1, \ldots, a_r\}$ from $\breve{R}$, then $R' = \breve{R}' \backslash \{a_1, \ldots, a_r\} \cup \{b\}$ meets the $\mathbb{C}$-superreduct condition. However, we know that b does not contribute to any $\mathbb{C}$-reduct. Hence, b is superfluous in R' because $R'' = R' \setminus \{b\}$ also meets the $\mathbb{C}$-superreduct condition. So, $\breve{R}' = \breve{R} \setminus \{b\}$ is also an r-$\mathbb{C}$-reduct and $|\breve{R}'| < |\breve{R}|$. Contradiction. □

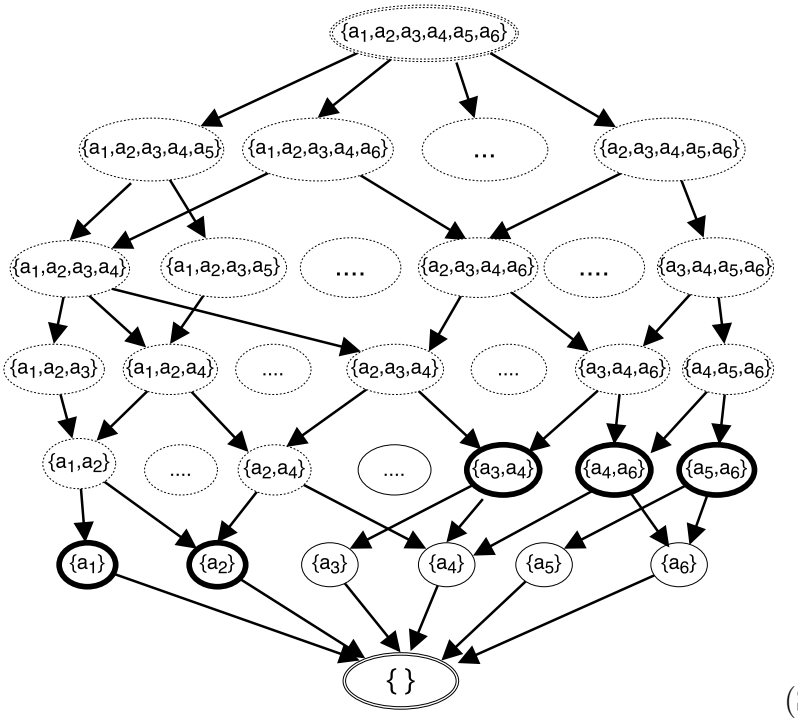
Remark 1. *(Zero redundancy)*
A 0-$\mathbb{C}$-reduct R is a $\mathbb{C}$-reduct.

Remark 2. *(Redundant attributes removal)*
After the removal of any attribute a from a non-empty r-$\mathbb{C}$-reduct $\breve{R}$, the remaining set $\breve{R}' = \breve{R} \setminus \{a\}$ satisfies the following:

1. *$\breve{R}'$ is a $(r-1)$-$\mathbb{C}$-superreduct*
2. *$\exists \breve{R}^{(r\text{-}1)} \subseteq \breve{R}'$, where $\breve{R}^{(r\text{-}1)}$ is a $(r-1)$-$\mathbb{C}$-reduct.*



(First iteration)



(Second iteration)

Figure 3.3: The interpretation of 'downward closure' property of Apriori in the case of reduct computation. The bold ovals correspond to reducts, meanwhile the dashed ovals correspond to sets that will not be explored, since in every iteration we remove all so-far-found superreducts from the F_k. Such an approach has a big impact on the amount of explored candidate sets and the overall algorithm performance.

3.3 Breadth First Search Algorithms

There are a lot of feature selection methods described in the literature (see Section 2.3), however it is hard to find those which would take into account not only the quality and relevance of selected attribute subsets but also their resilience to partial loss or lack of the data. Although there are some preliminary studies of algorithmic approaches that allow to construct resilient subsets of attributes basing on controlled redundancy in generated subsets [137], or relying on ensemble techniques [356], in general, it is not straightforward how to provide an expected resilience level to the feature selection.

In this section, we introduce a novel approach to perform resilient feature selection. It is inspired by the well-known Apriori algorithm that was adapted in many ways in both, RST-based [398] and non-RST-based [202] feature selection frameworks. The presented mechanism generates r-$\mathbb{C}$-reducts for a given implementation of a test function $test_C(R)$ for a criterion $\mathbb{C}$. The $test_C(R)$ allows us to verify whether a given subset of attributes R satisfies the examined criterion.

3.3.1 Apriori-based Algorithm

Originally, the Apriori algorithm was supposed to discover association rules between items in a database of sales transactions. Given a set of transactions, the problem is to generate all association rules that have support and confidence no less than the user-specified thresholds (called *minsup* and *minconf*, respectively). Apriori is characterized as a level-wise complete search (breadth first search) algorithm using anti-monotonicity of itemsets: "If an itemset is not frequent, any of its superset is never frequent", which is also called the downward closure.

For resilient attribute subsets, the downward closure property refers to the monotonicity of $\mathbb{C}$ (cf. Definition 5) – that is, if the subset R of attributes satisfies function $\mathbb{C}$, then every superset $\bar{R} \supseteq R$ does it too. Hence, we do not need to explore supersets of R. Moreover, for the optimization problems considered in this paper, it is enough to find minimal sets satisfying $\mathbb{C}$, so that the algorithm could stop (see Figure 3.3). Going further, in the case of resilient attribute subsets, we know that each r-$\mathbb{C}$-reduct may be reached only by adding an attribute to an $(r-1)$-$\mathbb{C}$-superreduct (see Remark 2).

The resilient version – r-*apriori_gen*(F_{k-1}) – of the original *apriori_gen* procedure, takes as an argument F_{k-1} – the set of all frequent $(k-1)$-items (in our case, attribute subsets of size $k-1$), and returns a superset 'C_k' containing all frequent k-itemsets. First, in the *'from'* part of the SQL implementation below, F_{k-1} is joined with all attributes from A (A may be represented as a single-column table with attributes in rows). In the *'group by'* phase, the set is compacted and some additional meta-data is created, e.g., *is-1-Superreduct, ..., is-r-Superreduct* properties are generated – which means that candidate R has particular resilience level r. Actually, one can compare this kind of SQL-based approach with some other Apriori-style SQL implementations [331], as well as SQL-based RST-related calculations [398].

The candidate set $R \in C_k$ is *is-r-Superreduct*, if and only if all subsets $R' \subset R$ that

$|R'| = |R| - 1$ are $r-1$ resilient (see Remark 2). Moreover, we rely on monotonicity of $test_{\mathbb{C}}$ – if any subset of candidate $R' \subset R$ satisfies $test_{\mathbb{C}}(R') = 1$ then R as well – this way, in some cases, we may omit necessity to perform $test_{\mathbb{C}}$ calculations. In the *'having'* part of SQL, we discard candidates corresponding to subsets that were left out earlier (*'count'* is less than k, if and only if some of subsets of R are missing in F_{k-1}). The *sort()* function is responsible for sorting attributes according to, e.g., a lexicographical order. Below is the SQL implementation of the *r-apriori_gen*(F_{k-1}):

```
INSERT into C_k
select sort(p.item_1, p.item_2, ..., p.item_{k-1}) as candidate,
       max(testC) as testC, min(testC) as is-1-Superreduct,
       min(is-1-Superreduct) as is-2-Superreduct, ...,
       min(is-(r-1)-Superreduct) as is-r-Superreduct, count(*)
from F_{k-1} p, A a
group by candidate
having count(*) = k
```

To better illustrate the proposed SQL implementation of *r-apriori_gen*, in Figure 3.4, we present two iterations of the procedure on the limited set A containing three attributes $A = \{a_1, a_2, a_3\}$. In the preliminary iteration (left snippet in Figure 3.4) it is necessary to apply $test_{\mathbb{C}}(a)$ for each attribute $a \in \{a_1, a_2, a_3\}$. The cost of such operation is $\mathcal{O}(|A|) \times \mathcal{O}(test_{\mathbb{C}})$. The result confirmed that attributes $\{a_1\}$ and $\{a_2\}$ satisfy $\mathbb{C}$ (let us call them $\mathbb{C}$-reducts), that is $test_{\mathbb{C}}(\{a_1\}) = test_{\mathbb{C}}(\{a_2\}) = 1$, however $\{a_3\}$ does not – $test_C(\{a_3\}) = 0$. In the first iteration (right snippet in Figure 3.4) there is no need to execute $test_{\mathbb{C}}$ at all, since all the sets: $\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$ has direct connection in the lattice to at least one $\mathbb{C}$-reduct, hence all satisfy $test_{\mathbb{C}} = 1$ because of the monotonicity of $\mathbb{C}$ (in the presented SQL implementation of *r-apriori_gen* it is interpreted as *'max(testC) as testC'*). Moreover, we know that the set $\{a_1, a_2\}$ is 1-$\mathbb{C}$-reduct, since every edge down in the lattice ends in a $\mathbb{C}$-reduct (interpreted as *'min(testC) as is-1-Superreduct'* in SQL). This short discussion shows that bottom-up approach based on *apriori_gen* allowed to conclude information about given superset basing only on properties of its subset, without necessity to perform additional calculations.

Algorithm 2 presents pseudo-code for r-apriori that, for the given r, generates all r-$\mathbb{C}$-superreducts or ends with minimal r-$\mathbb{C}$-reducts. The overall flow of r-apriori is almost the same as the original Apriori algorithm, however there are differences in the implementation of particular functions like, e.g., *r-apriori_gen*. In every iteration of the outer *'for'* loop in Algorithm 2, the candidate subsets are generated with the resilient *r-apriori_gen* procedure. The inner *'foreach'* loop iterates over generated candidates and verifies $test_{\mathbb{C}}$ function. F_k is built on candidate set without those sets, which are already recognized as r-$\mathbb{C}$-superreducts. Additionally, there are two flags $\{ALL_{\breve{R}}, MIN_{\breve{R}}\}$ that allow to control Algorithm 2 in order to generate all r-$\mathbb{C}$-reducts in $\mathbb{S}$ ($ALL_{\breve{R}}$), or just minimal ones ($MIN_{\breve{R}}$).

For a standard reduct problem, $\mathbb{C}$ may correspond to discernibility whereas $test_C(R)$ may correspond to function *isReduct* [398]. As other examples, $test_C$ may be implemented as correlation with a decision attribute, a constraint for conditional entropy (3.8), the RST-related function γ (3.4), and many others [139].

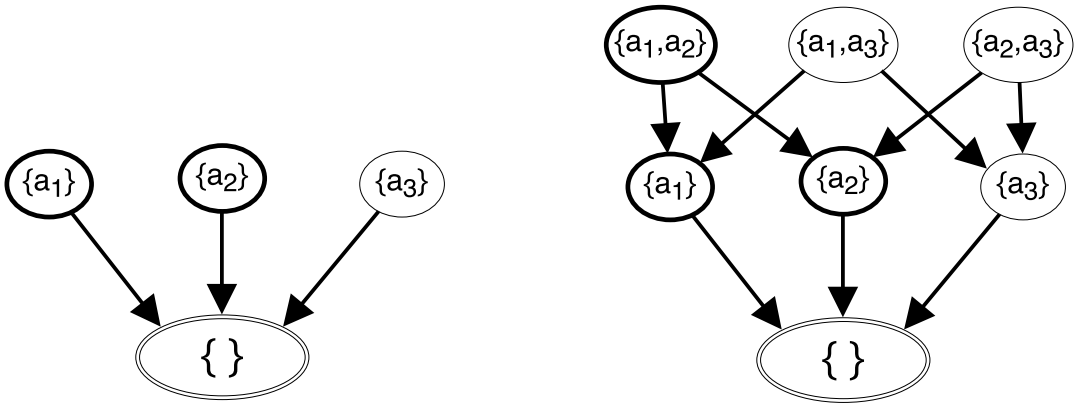


Figure 3.4: The preliminary (left) and the first (right) iteration of the *r-apriori_gen* procedure.

3.3.2 Algorithm Working Example

To confirm that Algorithm 2 generates all (smallest) r-$\mathbb{C}$-reducts (depending on the control flag) let us perform illustrative experiments for the data in Table 2.2. Let us refer to the conditional entropy H as the penalty measure and consider the criterion function $\mathbb{C}^{(H,\ 0.2)}$. The resilient version of $\mathbb{C}^{(H,\ 0.2)}$ may be constructed for a given r as described in equation (3.9). Let us now present a concise experiment for the data in Table 2.2, aimed at finding all / the smallest 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts.

In Table 3.1, we summarize the states after every iteration of *'for loop'* in Algorithm 2. In Table 3.2, each row corresponds to the call of the *r-apriori_gen* procedure, whereby the generated attribute subsets are assigned to one of four groups: *!testC, testC, is-1-Superreduct* and *is-2-Superreduct*. Note that for $k = 2$ (the second iteration of *r-apriori_gen*) minimal 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts were found, with candidate sets $\{a_1, a_2\}$, $\{a_1, a_6\}$, $\{a_2, a_6\}$ assigned to *is-1-Superreduct* group. Thus, in the case of *'Flag'* set to $MIN_{\breve{R}}$, the algorithm would stop its execution (compare the if condition in line 12).

3.4 Computational Complexity Study

A natural question arises whether the most meaningful, particularly minimal r-$\mathbb{C}$-reducts can be derived from the data in a more efficient way than by using the aforementioned breath first search techniques. Intuitively, having in mind the already-published NP-hardness results corresponding to minimal r-reducts [137] – somewhat similar to (multi)set multicover problems studied, e.g., in [167] – we should not expect the existence of fast deterministic algorithms with this respect. Still, one might think that this kind of complexity could depend on the choice of criterion function, i.e., although the problem of finding minimal r-reducts (Definition 8) is known to be NP-hard, the analogous problems of finding minimal r-$\mathbb{C}$-reducts (Definition 9) could be computationally more feasible at least for some functions $\mathbb{C}$.

In this section, we prove that every NP-hard attribute reduction problem P

Algorithm 2: r-apriori for resilient feature selection

Data: F_{k-1}; $\mathbb{S} = (U, A \cup \{d\})$; $Flag \in \{ALL_{\breve{R}}, MIN_{\breve{R}}\}$
Result: $\mathbb{R}$ – a set of all r-$\mathbb{C}$-reducts or all smallest r-$\mathbb{C}$-reducts

```
k ← 1; r ← 0; F_k ← A; ℝ ← ∅
for k = 2; F_{k−1} ≠ ∅; k++ do
    /* Generate candidates with r-apriori_gen */
    C_k ← r-apriori_gen(F_{k−1})
    foreach candidate R ∈ C_k do
        if !R.test_C then
            R.test_C ← test_C(R)
        end
    end
    ℛ_r ← {R ∈ C_k | R is r-ℂ-superreduct}
    ℝ ← ℝ ∪ ℛ_r
    if Flag = MIN_Ř AND |ℝ| > 0 then
        break;
    end
    F_k ← C_k \ ℛ_r
end
return ℝ
```

Table 3.1: Summary of experiments for r-Apriori.

k	$\mathcal{R}_r$	$\mathbb{R}$	comments
1	$\emptyset$	$\emptyset$	preliminary step; k=1
2	$\{a_1, a_2\}$, $\{a_1, a_6\}$, $\{a_2, a_6\}$	$\{a_1, a_2\}$, $\{a_1, a_6\}$, $\{a_2, a_6\}$	minimal 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts
3	$\{a_1, a_4, a_5\}$, $\{a_2, a_4, a_5\}$, $\{a_3, a_4, a_5\}$, $\{a_3, a_4, a_6\}$	$\{a_1, a_2\}$, $\{a_1, a_6\}$, $\{a_2, a_6\}$, $\{a_1, a_4, a_5\}$, $\{a_2, a_4, a_5\}$, $\{a_3, a_4, a_5\}$, $\{a_3, a_4, a_6\}$	all 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts

that may be expressed via an appropriately defined criterion function $\mathbb{C}^P$ remains NP-hard even for its resilient version $\breve{P}$, where r refers to the resilience level and means that any r attributes of the examined set R may be unavailable without any impact on the criterion $\mathbb{C}^P$. The presented NP-hardness proof mechanism works for any functions $\mathbb{C}$ that meet the requirements of Definition 5. On the one hand, one may say that it overlaps with other complexity studies. For instance, let us refer to NP-hardness of partial multi-cover problems [314], which might be used as a prerequisite for proving NP-hardness of a resilient version of the aforementioned minimal α-reduct problem [276]. On the other hand, our theoretical result is broader as it allows us to deal with a far wider family of formulations of attribute reduction problems [351, 367].

Table 3.2: The course of experiments for *r-apriori_gen*.

k	C_k	
1	!testC	$\{a_3\}$, $\{a_4\}$, $\{a_5\}$
	testC	$\{a_1\}$, $\{a_2\}$, $\{a_6\}$
	is-1-Superreduct	$\emptyset$
	is-2-Superreduct	$\emptyset$
2	!testC	$\emptyset$
	testC	$\{a_1, a_3\}$, $\{a_1, a_4\}$, $\{a_1, a_5\}$, $\{a_2, a_3\}$, $\{a_2, a_4\}$, $\{a_2, a_5\}$, $\{a_3, a_6\}$, $\{a_4, a_6\}$, $\{a_5, a_6\}$, $\{a_3, a_4\}$, $\{a_3, a_5\}$, $\{a_4, a_5\}$
	is-1-Superreduct	$\{a_1, a_2\}$, $\{a_1, a_6\}$, $\{a_2, a_6\}$
	is-2-Superreduct	$\emptyset$
3	!testC	$\emptyset$
	testC	$\emptyset$
	is-1-Superreduct	$\{a_1, a_4, a_5\}$, $\{a_2, a_4, a_5\}$, $\{a_3, a_4, a_5\}$, $\{a_3, a_4, a_6\}$
	is-2-Superreduct	$\{a_1, a_2, a_6\}$

3.4.1 *A*-Attributes

For further needs, let us consider a family of artificial attributes, so-called A-attributes, denoted as $\#attr$. The values of $\#attr$ are constructed, for each $u \in U$, as concatenations of all values $a(u)$, $a \in A$[1]. Polynomial reduction presented later in Subsection 3.4.2 is based on the following properties of A-attributes.

Lemma 1 (A-attribute $\#attr$)**.**
For a given decision table $\mathbb{S} = (U, A \cup \{d\})$ and $\mathbb{C} : \mathcal{P}(A) \to \{0, 1\}$ we may generate an arbitrary number of A-attributes $\#attr$ such that:

1. *For any $n \in \mathbb{N}$, for all i, j such that $1 \leq i, j \leq n$, there is $\mathbb{C}(\{\#attr_i\}) = \mathbb{C}(\{\#attr_i, \#attr_{i+1}, \ldots, \#attr_j\}) = \mathbb{C}(A) = \mathbb{C}(A \cup \{\#attr_i, \#attr_{i+1}, \ldots, \#attr_j\})$.*

2. *Singleton sets $\{\#attr_i\}$ are the smallest non-empty subsets of attributes satisfying $\mathbb{C}$ in the extended decision table $\mathbb{S}' = (U, A \cup \{\#attr_{1 \leq i \leq n}\} \cup \{d\})$.*

Proof. Item 1: Since every $\#attr$ is generated as concatenation of all attribute values, there are the following functional dependencies ($\Rrightarrow$) between A and each $\#attr$: For any $n \in \mathbb{N}$, for every i, j, such that $1 \leq i \leq n$, there is $A \Rrightarrow \#attr_i \wedge \#attr_i \Rrightarrow A$. So, $\mathbb{C}(\{\#attr\}) \leq \mathbb{C}(A) \wedge \mathbb{C}(\{\#attr\}) \geq \mathbb{C}(A)$. Thus, given $1 \leq i, j \leq n$, $\mathbb{C}(\{\#attr\}) = \mathbb{C}(\{\#attr_i, \#attr_{i+1}, \ldots, \#attr_j\}) = \mathbb{C}(A) = \mathbb{C}(A \cup \{\#attr_i, \#attr_{i+1}, \ldots, \#attr_j\})$.

Item 2: A singleton set $\{\#attr\}$ satisfies $\mathbb{C}$ (that is: $\mathbb{C}(\{\#attr\}) = 1$) and $\{\#attr\}$ is a single attribute, hence it is the smallest one – what ends the proof. □

Let us strengthen the meaning of Lemma 1 with the following remarks:

[1] If attribute domains are overlapping, i.e., there exist $a_i, a_j \in A$ for which $V_{a_i} \cap V_{a_j} \neq \emptyset$, then concatenation may include a delimiter $|_A$ such that for each $a \in A$ we have $|_A \notin V_a$.

Lemma 2 (Only one $\mathbb{C}$-reduct contains $\#attr$)**.**
If $\mathbb{C}(\emptyset) \neq \mathbb{C}(A)$, then, for each i, $R = \{\#attr_i\}$ is the smallest $\mathbb{C}$-superreduct in decision table $\mathbb{S}' = (U, A \cup \{\#attr_{1 \leq i \leq n}\} \cup \{d\})$ and, in particular, there are no other $\mathbb{C}$-reducts containing $\#attr_i$.

Proof. Straightforward. In particular, we know that $\{\#attr_i\}$ is a $\mathbb{C}$-reduct. Therefore, for any set $R = \{\#attr_i, a\}$, a is dispensable. □

Lemma 3 ($\#ATTRs$ forms the smallest r-$\mathbb{C}$-reduct)**.**
If $\mathbb{C}(\emptyset) \neq \mathbb{C}(A)$, then the set $\#ATTRs = \{\#attr_1, \ldots, \#attr_r, \#attr_{r+1}\}$ is the smallest set of attributes that satisfies r-$\mathbb{C}$ in $\mathbb{S}' = (U, A \cup \{\#attr_{1 \leq i \leq n}\} \cup \{d\})$.

Proof. We need to show the two following things. First, $\#ATTRs$ is an r-$\mathbb{C}$-reduct. Second, it is the smallest one.

1. After removal of any r elements from $\#ATTRs$ there is still one $\#attr$ attribute left. From Lemma 1, we know that such attribute satisfies $\mathbb{C}$. Hence, $\#ATTRs$ is the r-$\mathbb{C}$-superreduct.

2. Assume that there is $\breve{R}'$ that satisfies r-$\mathbb{C}$ and $|\breve{R}'| \leq r < r + 1 = |\#ATTRs|$. If so, after removal of r attributes from $\breve{R}'$ there is no attribute. Since $\mathbb{C}(\emptyset) \neq \mathbb{C}(A)$, i.e., $\mathbb{C}(\emptyset) = 0$ and $\mathbb{C}(A) = 1$, $\breve{R}'$ is not an r-$\mathbb{C}$-reduct. Contradiction.

□

Lemma 4 (Reducts and $\#ATTRs$)**.**
Let R be a non-empty $\mathbb{C}$-reduct in $\mathbb{S}' = (U, A \cup \{\#attr_{1 \leq i \leq n}\} \cup \{d\})$, such that $R \cap \{\#attr_1, \ldots, \#attr_r\} = \emptyset$. Then, the set $\breve{R} = R \cup \{\#attr_1, \ldots, \#attr_r\}$ is an r-$\mathbb{C}$-reduct.

Proof. From Lemma 3 we know that $\{\#attr_1, \ldots, \#attr_r\}$ is the smallest $(r-1)$-$\mathbb{C}$-reduct. From Lemma 2 we know that for each $\mathbb{C}$-reduct $R \subset A$ and $\#attr$, if $R \neq \{\#attr\}$, then $R \cap \{\#attr\} = \emptyset$. To show that $\breve{R} = R \cup \{\#attr_1, \ldots, \#attr_r\}$ is an r-$\mathbb{C}$-reduct, we need to prove that for any $R' \subset \breve{R}$ such that $|R'| \leq r$ the remaining set $\breve{R} \setminus R'$ is a $\mathbb{C}$-superreduct, thus it satisfies the condition $\mathbb{C}(\breve{R} \setminus R') = 1$.

There are two cases to be considered. First, $R' = \{\#attr_1, \ldots, \#attr_r\}$. In that case the remaining set after $\breve{R} \setminus R'$ is R. Hence, it is a $\mathbb{C}$-reduct. Second, $R' \neq \{\#attr_1, \ldots, \#attr_r\}$. In that case the remaining set $\breve{R} \setminus R'$ contains at least one $\#attr$ attribute. So, from Lemma 1 it satisfies $\mathbb{C}(\breve{R} \setminus R') = 1$. Thus, it is a $\mathbb{C}$-superreduct. □

Let us continue with the study of the impact of $\#attr$ attributes on the properties of $\mathbb{S}$ and $\mathbb{C}$, with an emphasis on the extended data representation $\mathbb{S}'$. In order to make a proper distinction between $\mathbb{S}$ and $\mathbb{S}'$, we go back to the aforementioned explicit data-specific notation $\mathbb{C}_{\mathbb{S}}$ and $\mathbb{C}_{\mathbb{S}'}$, respectively.

Lemma 5 (Impact of $\#attr$ on $\mathbb{S}$ and $\mathbb{C}$)**.**
Let $\mathbb{S} = (U, A \cup \{d\})$ be a decision table, $\mathbb{C} : \mathcal{P}(A) \to \{0, 1\}$ be a given criterion. Let $\mathbb{S}' = (U, A \cup \{\#attr_1, \ldots, \#attr_r\} \cup \{d\})$ be an extended data representation. Then, the following properties hold:

1. $\forall_{R\subseteq A}$ *if* $\mathbb{C}_{\mathbb{S}}(R) = 0$ *then* $\mathbb{C}_{\mathbb{S}'}(R) = 0$;
2. $\forall_{R\subseteq A}$ *if* $\mathbb{C}_{\mathbb{S}}(R) = 1$ *then* $\mathbb{C}_{\mathbb{S}'}(R) = 1$;
3. $\forall_{R\subseteq A}\ \mathbb{C}_{\mathbb{S}}(R) = \mathbb{C}_{\mathbb{S}'}(R)$
4. *If* $\forall_{R'\subseteq A\cup\{\#attr_1,\ldots,\#attr_r\}}\ \mathbb{C}_{\mathbb{S}'}(R') = 0$ *then* $\forall_{R\subseteq A}\mathbb{C}_{\mathbb{S}}(R) = 0$;
5. *If* $\exists_{R'\subseteq A\cup\{\#attr_1,\ldots,\#attr_r\}}\ \mathbb{C}_{\mathbb{S}'}(R') = 1$ *then* $\exists_{R\subseteq A}\mathbb{C}_{\mathbb{S}'}(R) = \mathbb{C}_{\mathbb{S}}(R) = 1$.

Proof. Item 1: If $\mathbb{C}_{\mathbb{S}}(A) = 0$ then from Lemma 1 $\mathbb{C}_{\mathbb{S}}(A) = \mathbb{C}_{\mathbb{S}'}(A \cap \{\#attr_1, \ldots, \#attr_r\}) = 0$. So, from monotonicity $\mathbb{C}_{\mathbb{S}'}(R) = 0$. If $\mathbb{C}_{\mathbb{S}}(A) = 1$ then $\mathbb{C}_{\mathbb{S}}(R) < \mathbb{C}_{\mathbb{S}}(A)$. From Lemma 1 we have $\mathbb{C}_{\mathbb{S}}(A) = \mathbb{C}_{\mathbb{S}'}(A \cap \{\#attr_1, \ldots, \#attr_r\})$. So, $\mathbb{C}_{\mathbb{S}'}(R) < \mathbb{C}_{\mathbb{S}'}(A \cap \{\#attr_1, \ldots, \#attr_r\})$.

Item 2: Directly from Lemma 1.

Item 3: Directly from items (1) and (2).

Item 4: By contradiction. We have that for each $R' \subseteq A \cup \{\#attr_1, \ldots, \#attr_r\}$ there is $\mathbb{C}_{\mathbb{S}'}(R') = 0$. Let $R \subseteq A$ and $\mathbb{C}_{\mathbb{S}}(R) = 1$. Then, directly from (2), $\mathbb{C}_{\mathbb{S}'}(R) = 1$.

Item 5: If $R \cap \{\#attr_1, \ldots, \#attr_r\} = \emptyset$ then $R' \subseteq A$. Otherwise, if $R \cap \{\#attr_1, \ldots, \#attr_r\} \neq \emptyset$ then from Lemma 1, we have $\mathbb{C}_{\mathbb{S}}(A) = \mathbb{C}_{\mathbb{S}'}(A) = \mathbb{C}_{\mathbb{S}'}(\{\#attr_1, \ldots, \#attr_r\}) = \mathbb{C}_{\mathbb{S}'}(A \cup \{\#attr_1, \ldots \#attr_r\}) = 1$. Thus, A is the solution, i.e., $\mathbb{C}_{\mathbb{S}}(A) = \mathbb{C}_{\mathbb{S}'}(A) = 1$.

□

3.4.2 Resilient NP-hardness

In this section, we concentrate on showing that every NP-hard attribute reduction problem P expressed by means of criterion function $\mathbb{C}^P$ remains NP-hard also for its resilient variant $\breve{P}$ expressed by r-$\mathbb{C}^P$.

Theorem 1. *(NP-hardness of minimal r-$\mathbb{C}^P$)*
Let P be a problem of finding the minimal set R satisfying condition expressed via a criterion $\mathbb{C}^P : \mathcal{P}(A) \to \{0, 1\}$. If P is NP-hard, then finding minimal set $\breve{R}$ satisfying r-$\mathbb{C}^P$ (see equation (3.9)) is also NP-hard.

To prove Theorem 1, we will show that P can be polynomially reduced to $\breve{P}$. The reduction is as follows. Given a problem input, i.e., a decision table $\mathbb{S} = (U, A \cup \{d\})$, the reduction comes to creation of a new data representation $\mathbb{S}'$ that contains additional r conditional A-attributes, where r corresponds to the expected resilience level. Obviously, the whole reduction is polynomial:

1. Given the original data representation $\mathbb{S} = (U, A \cup \{d\})$ and integer r
2. Add r $\#attr$ attributes and, this way, create the extended data representation $\mathbb{S}' = (U,\ A \cup \{\#attr_1, \ldots, \#attr_r\} \cup \{d\})$
3. Solve the problem $\breve{P}$ defined via r-$\mathbb{C}^P$ (3.9)
4. Extract the solution of P as described below.

To show that the reduction is applicable, we need to show two things. First, if there is a solution of P, then there must be a solution of $\breve{P}$ ($P \rightarrow \breve{P}$). Second, if there is a solution of $\breve{P}$, then there must be a solution of P ($\breve{P} \rightarrow P$).

In order to distinguish applications of functions $\mathbb{C}^P$ and r-$\mathbb{C}^P$ on the original decision table $\mathbb{S}$ and the extended data representation $\mathbb{S}$', we use the following notation: $\mathbb{C}_{\mathbb{S}}$, r-$\mathbb{C}_{\mathbb{S}}$, $\mathbb{C}_{\mathbb{S}'}$ and r-$\mathbb{C}_{\mathbb{S}'}$, respectively.

$P \rightarrow \breve{P}$. First, let us discuss the boundary condition $\mathbb{C}_{\mathbb{S}}(\emptyset) = \mathbb{C}_{\mathbb{S}}(A)$: a) If $\mathbb{C}_{\mathbb{S}}(A) = 0$ there is no solution of P in $\mathbb{S}$. So, from Lemma 1 we have $\mathbb{C}_{\mathbb{S}'}(A \cup \#ATTRs) = 0$. So, there is no solution of P in $\mathbb{S}$'. Thus, the equation (3.9) is never met. Hence, there is no solution of $\breve{P}$ in $\mathbb{S}$'. b) If $\mathbb{C}_{\mathbb{S}}(\emptyset) = 1$ everything is a solution of P and $\mathbb{C}_{\mathbb{S}'}(\emptyset) = 1$ – directly from Lemma 5. The equation (3.9) is always met so r-$\mathbb{C}_{\mathbb{S}'}(\emptyset) = 1$. Hence, everything is a solution of $\breve{P}$ in $\mathbb{S}$'. Below, we consider the more complex case when $\mathbb{C}_{\mathbb{S}}(\emptyset) \neq \mathbb{C}_{\mathbb{S}}(A)$:

Let R, $|R| = k$, be a solution of P, in particular, $\mathbb{C}_{\mathbb{S}}(R) = 1$. Let $\#ATTRs = \{\#attr_1, \ldots, \#attr_r\}$ be a set of r $\#attr$ attributes $|\#ATTRs| = r$. Let $\breve{R} = R \cup \#ATTRs$. Directly from Lemma 4, we have r-$\mathbb{C}_{\mathbb{S}'}(R \cup \#ATTRs) = 1$.

Now, we need to show that $\breve{R} = R \cup \#ATTRs$ of size $|\breve{R}| = k + r$ is minimal. Assume that there exists $\breve{R}' \subseteq A \cup \{\#attr_1, \ldots, \#attr_r\}$ such that r-$\mathbb{C}_{\mathbb{S}'}(\breve{R}') = 1$ and $|\breve{R}'| < |\breve{R}|$. From equation (3.9), we know that after removing any r attributes from $\breve{R}'$ we have R' of size $l < k$ ($|R'| < |R|$) that satisfies $\mathbb{C}_{\mathbb{S}'}$ (in $\mathbb{S}$'). From Lemma 5, we know that R' satisfies also $\mathbb{C}_{\mathbb{S}}$ (in $\mathbb{S}$). Whereas, R of size k is a solution of P for $\mathbb{S}$. Contradiction. □

In order to prove $\breve{P} \rightarrow P$, let us introduce an auxiliary lemma for $\#ATTRs$:

Lemma 6 ($\#ATTRs$ and $\breve{P}$)**.**
If there is a non-empty solution $\breve{R}$ to $\breve{P}$ in $\mathbb{S}' = (U, A \cup \{\#attr_1, \ldots, \#attr_r\} \cup \{d\})$, then there exists a solution $\breve{R}'$ to $\breve{P}$ in $\mathbb{S}'$ that satisfies:

1. $|\breve{R}'| = |\breve{R}|$
2. $\{\#attr_1, \ldots, \#attr_r\} \subseteq \breve{R}'$

As a proof, we present a constructive algorithm that transforms a given solution $\breve{R}$ to the problem $\breve{P}$ into $\breve{R}'$, whereby $|\breve{R}'| = |\breve{R}|$ and $\{\#attr_1, \ldots, \#attr_r\} \subseteq \breve{R}'$. Let $\breve{R}$ be a solution to $\breve{P}$ in $\mathbb{S}' = (U, A \cup \{\#attr_1, \ldots, \#attr_r\} \cup \{d\})$.

In the first step, we remove all $\#attr$ attributes from $\breve{R}$, where $|\breve{R} \cap \{\#attr_1, \ldots, \#attr_r\}| = m$ and $0 \leq m \leq r$. Next, we remove any $r - m$ other attributes. The remaining set R of size $|\breve{R}| - r$ is satisfying $\mathbb{C}_{\mathbb{S}'}(R) = 1$ (see equation (3.9)).

In the second step, we create a solution $\breve{R}' = R \cup \{\#attr_1, \ldots, \#attr_r\}$. We know that the solution $\breve{R}'$ constructed this way satisfies r-$\mathbb{C}_{\mathbb{S}'}(\breve{R}') = 1$ and is minimal because $|\breve{R}'| = |\breve{R}|$. The complexity of the above algorithm is obviously polynomial.

$\breve{P} \rightarrow P$. First, let us discuss the case of $\mathbb{C}_{\mathbb{S}'}(\emptyset) = \mathbb{C}_{\mathbb{S}'}(A)$: a) If r-$\mathbb{C}_{\mathbb{S}'}(A \cup \{\#attr_1, \ldots, \#attr_r\}) = 0$ there is no solution of $\breve{P}$ in $\mathbb{S}$'. Hence, from equation (3.9), for any $R \subseteq A \cup \{\#attr_1, \ldots, \#attr_r\}$, if $|R| \geq r$, then $\mathbb{C}_{\mathbb{S}'}(A \cup \{\#attr_1, \ldots, \#attr_r\} \setminus R) = 0$. Thus, for $R = \{\#attr_1, \ldots, \#attr_r\}$ we have $\mathbb{C}_{\mathbb{S}'}(A) = 0$. So, from monotonicity of

Table 3.3: An exemplary decision table extended with two A-attributes.

$U \backslash A$	a_1	a_2	a_3	a_4	a_5	a_6	$\#attr_1$	$\#attr_2$	d
u_1	f	'a'	$\bigtriangledown$	$\circ$	$\square$	'x'	f'a'$\bigtriangledown \circ \square$'x'	f'a'$\bigtriangledown \circ \square$'x'	*good*
u_2	f	'b'	$\bigtriangledown$	$\odot$	$\bullet$	'x'	f'b'$\bigtriangledown \odot \bullet$'x'	f'b'$\bigtriangledown \odot \bullet$'x'	*good*
u_3	f	'c'	$\bigtriangleup$	$\oslash$	$\diamond$	'x'	f'c'$\bigtriangleup \oslash \diamond$'x'	f'c'$\bigtriangleup \oslash \diamond$'x'	*good*
u_4	f	'd'	$\bigtriangledown$	$\otimes$	$\triangleleft$	'x'	f'd'$\bigtriangledown \otimes \triangleleft$'x'	f'd'$\bigtriangledown \otimes \triangleleft$'x'	*good*
u_5	f	'e'	$\bigtriangledown$	$\ominus$	$\star$	'y'	f'e'$\bigtriangledown \ominus \star$'y'	f'e'$\bigtriangledown \ominus \star$'y'	*good*
u_6	f	'f'	$\bigtriangledown$	$\oslash$	$\triangleright$	'y'	f'f'$\bigtriangledown \oslash \triangleright$'y'	f'f'$\bigtriangledown \oslash \triangleright$'y'	*good*
u_7	t	'g'	$\bigtriangleup$	$\otimes$	$\square$	'y'	t'g'$\bigtriangleup \otimes \square$'y'	t'g'$\bigtriangleup \otimes \square$'y'	*bad*
u_8	t	'h'	$\bigtriangleup$	$\ominus$	$\bullet$	'z'	t'h'$\bigtriangleup \ominus \bullet$'z'	t'h'$\bigtriangleup \ominus \bullet$'z'	*bad*
u_9	t	'i'	$\bigtriangledown$	$\oplus$	$\diamond$	'z'	t'i'$\bigtriangledown \oplus \diamond$'z'	t'i'$\bigtriangledown \oplus \diamond$'z'	*bad*

$\mathbb{C}$, we have that for any $R \subseteq A$ there is $\mathbb{C}_{\mathbb{S}}(A) = 0$. b) If r-$\mathbb{C}_{\mathbb{S}'}(\emptyset) = 1$ then from equation (3.9) $\mathbb{C}_{\mathbb{S}'}(\emptyset) = 1$. Hence, from Lemma 5 $\mathbb{C}_{\mathbb{S}}(\emptyset) = 1$.

Now, we consider the case of $\mathbb{C}_{\mathbb{S}}(\emptyset) \neq \mathbb{C}_{\mathbb{S}}(A)$. Let $\breve{R}$ be a solution to $\breve{P}$ in $\mathbb{S}' = (U, A \cup \{\#attr_1, \ldots, \#attr_r\} \cup \{d\})$. Without loosing generality, we may assume that $\{\#attr_1, \ldots, \#attr_r\} \subseteq \breve{R}$ (see Lemma 6). Now, we must show that $R = \breve{R} \setminus \{\#attr_1, \ldots, \#attr_r\}$ is a solution of P in $\mathbb{S} = (U, A \cup \{d\})$.

From equation (3.9), we know that after removing any r attributes from $\breve{R}$ the remaining set R satisfies $\mathbb{C}_{\mathbb{S}'}(R) = 1$. Hence, it is a solution to P in $\mathbb{S}'$. From Lemma 5, we know that $\mathbb{C}_{\mathbb{S}}(R) = \mathbb{C}_{\mathbb{S}}(A) = 1$. So R is a solution to P in $\mathbb{S}$.

The last thing is to show that $R = \breve{R} \setminus \{\#attr_1, \ldots, \#attr_r\}$ constructed this way is minimal in $\mathbb{S}$. Suppose that there is a solution $R' \subseteq A$ that $\mathbb{C}_{\mathbb{S}}(R') = 1$ in $\mathbb{S}$ and $|R'| < |R|$. Thus, we may construct set $\breve{R}' = R' \cup \{\#attr_1, \ldots, \#attr_r\}$ that satisfies r-$\mathbb{C}_{\mathbb{S}'}(\breve{R}') = 1$ and $|\breve{R}'| < |\breve{R}|$. Contradiction, because $\breve{R}$ is minimal in $\mathbb{S}$', what ends the proof. □

3.4.3 Visual Interpretation

In order to provide a better understanding of the proof of Theorem 1, let us present a visual interpretation of the aforementioned reduction. In Table 3.3, we may find the exemplary decision table extended with two additional A-attributes created as described in Subsection 3.4.1. That is, for each object $u \in U$, we put $\#attr_1(u) = \#attr_2(u) = concat(a_1(u), a_2(u), a_3(u), a_4(u), a_5(u), a_6(u))$.

In the lattice in Figure 3.5, we can see the additional A-attributes. For simplicity, let us consider the most standard case of (r-)reducts introduced in Definitions 1 and 8. We remember that each #attribute forms a row identifier that maintains the same functional dependencies with the decision attribute d as the full attribute set $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$. Hence, singleton sets $\{\#attr_1\}$ and $\{\#attr_2\}$ are decision reducts. (As we know, for the considered data set the empty set of attributes is not a reduct.) Furthermore, once we consider attribute sets presented above *R1* line in Figure 3.5, we may notice that every two-element combination of attributes a_1, a_2, $\#attr_1$, $\#attr_2$ is a 1-reduct because after removal of any attribute from $\{a_1, a_2\}$, $\{a_1, \#attr_1\}$, $\{a_1, \#attr_2\}$, $\{\#attr_1, \#attr_2\}$ the remaining singletons constitute reducts. Similarly, every combination of three attributes out of a_1, a_2, $\#attr_1$, $\#attr_2$ is a

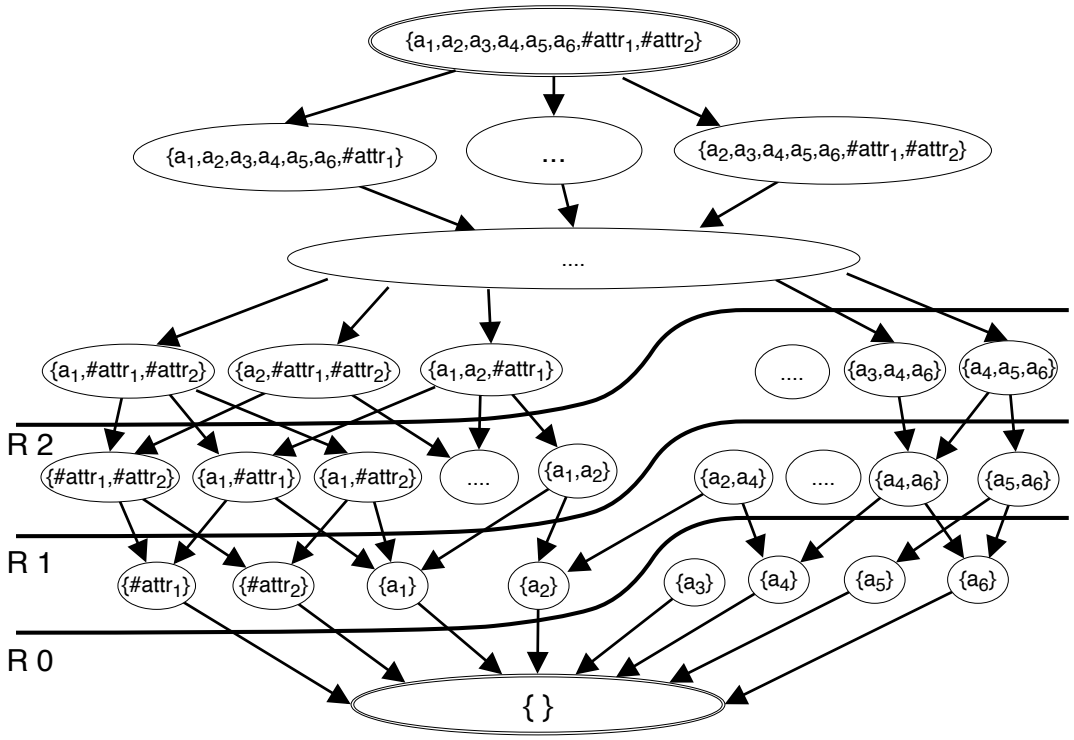


Figure 3.5: The lattice for the data in Table 3.3 with two A-attributes.

2-reduct, etc.

Let us discuss briefly the polynomial reduction basing on an exemplary visualization of decision table (2.2). In the lattice in Figure 2.2, we may easily notice that there are two minimal reducts: $\{a_1\}$ and $\{a_2\}$ in the original decision table (2.2). A presented reduction is as follows: in the first step, we add two A-attributes, this way we create extended decision table (3.3) - obviously the computational complexity of this step is polynomial with respect to the original decision table size. In Figure 3.5, we present the impact of the aforementioned extension on the lattice.

Let us assume that there is a polynomial algorithm $\breve{Alg}^2()$ that solves the minimal 2-reduct problem. Hence, the result of Algorithm $\breve{Alg}^2()$ executed on the decision table (3.3) is one of the following sets: $\{a_1,\ a_2,\ \#attr_1\}$, $\{a_1,\ a_2,\ \#attr_2\}$, $\{a_1,\ \#attr_1,\ \#attr_2\}$, $\{a_2,\ \#attr_1,\ \#attr_2\}$ - without losing generality let it be $\{a_1,\ a_2,\ \#attr_2\}$. According to the reduction presented in Subsection 3.4.2, we should remove all A-attributes from the resulting set, this way we have: $\{a_1,\ a_2\}$. Now, we may remove any attribute. Without loosing generality let it be a_1. We may easily notice that the remaining set $\{a_2\}$ is a decision reduct in the original decision table (2.2).

3.4.4 Impact of Complexity Study

Let us briefly discuss the impact of Theorem 1. For example, consider the resilient version of the minimal (H,ε)-approximate reduct problem or equivalently minimal $\mathbb{C}^{(H,\varepsilon)}$-reduct problem – using the nomenclature of equation (3.8). We may define r-$\mathbb{C}^{(H,\varepsilon)}$ as shown in equation (3.9). We obtain the following:

Theorem 2. *(Minimal r-$\mathbb{C}^{(H,\varepsilon)}$-reduct problem is NP-hard)*
For each $\varepsilon \in [0,1)$ and $r \in \mathbb{N}$, the minimal r-$\mathbb{C}^{(H,\varepsilon)}$-reduct problem – i.e., the problem

of finding, for an input decision table $\mathbb{S}$ *the minimal (in the sense of the number of elements) subset* $\breve{R} \subseteq A$ *such that* $r\text{-}\mathbb{C}^{(H,\varepsilon)}(\breve{R}) = 1$ *– is NP-hard.*

Proof. From Proposition 3.1.4, we know that $\mathbb{C}^{(H,\varepsilon)}$ satisfies conditions of Definition 5. As already mentioned, the minimal (H,ε)-approximate reduct problem is NP-hard [352]. Thus, directly from Theorem 1, the minimal $r\text{-}\mathbb{C}^{(H,\varepsilon)}$-reduct problem is NP-hard too. □

Similarly, we may refer to the minimal α-reduct problem [276], which – as already discussed – corresponds to the *Disc* measure (2.5).

Theorem 3. *(Minimal* $r\text{-}\mathbb{C}^{(Disc,\varepsilon)}$*-reduct problem is NP-hard)*
For each $\varepsilon \in [0,1)$ *and* $r \in \mathbb{N}$*, the minimal* $r\text{-}\mathbb{C}^{(Disc,\varepsilon)}$*-reduct problem – i.e., the problem of finding, for an input decision table* $\mathbb{S}$ *the minimal (in the sense of the number of elements) subset* $\breve{R} \subseteq A$ *such that* $r\text{-}\mathbb{C}^{(Disc,\varepsilon)}(\breve{R}) = 1$ *– is NP-hard.*

Proof. From Proposition 3.1.2, we know that $\mathbb{C}^{(Disc,\varepsilon)}$ satisfies conditions of Definition 5. Since we know that the minimal α-reduct problem is NP-hard [276] and the criterion $r\text{-}\mathbb{C}^{(Disc,\varepsilon)}(\breve{R}) = 1$ is equivalent for the one formulated for α-reducts for $\alpha = 1-\varepsilon$, thus from Theorem 1 we know that the minimal $r\text{-}\mathbb{C}^{(Disc,\varepsilon)}$-reduct problem is also NP-hard. □

The last example of the complexity result derivable directly from Theorem 1 is the following. However, let us note that the same mechanism could be applied for many other cases as well, in particular, for any formulations of $\mathbb{C}^{(Q,\varepsilon)}$-reducts for which the corresponding measures Q satisfy conditions of Definition 7 [351, 353].

Theorem 4. *(Minimal* $r\text{-}\mathbb{C}^{(\gamma,\varepsilon)}$*-reduct problem is NP-hard)*
For each $\varepsilon \in [0,1)$ *and* $r \in \mathbb{N}$*, the minimal* $r\text{-}\mathbb{C}^{(\gamma,\varepsilon)}$*-reduct problem – i.e., the problem of finding, for an input decision table* $\mathbb{S}$ *the minimal (in the sense of the number of elements) subset* $\breve{R} \subseteq A$ *such that* $r\text{-}\mathbb{C}^{(\gamma,\varepsilon)}(\breve{R}) = 1$ *– is NP-hard.*

Proof. From Proposition 3.1.2, we know that $\mathbb{C}^{(\gamma,\varepsilon)}$ satisfies conditions of Definition 5. Since the minimal (γ,ε)-approximate reduct problem is NP-hard [367], thus from Theorem 1 we know that the minimal $r\text{-}\mathbb{C}^{(\gamma,\varepsilon)}$-reduct problem is also NP-hard. □

3.5 Depth First Search Algorithms

The task of heuristic search of reducts in a given data set is broadly investigated in the literature. For instance, in [356] a combination of iterative filter-based feature selection algorithm with a statistical significance stop criterion and an RST-based redundant feature elimination was applied. A similar mixture of iterative feature selection and reduction was suggested in [80]. In [412], three groups of algorithms based on the deletion, addition-deletion and addition strategies were discussed. There are also many other approaches to algorithmic reduct construction that refer, e.g., to randomized search [98], ensembles [397], or various methods of feature granulation [139].

In the following subsections, we consider two approaches to the depth first search exploration of the lattice, which allow us to identify subsets of attributes that satisfy

Algorithm 3: *r-test*$_{\mathbb{C}}$

Data: $\breve{R}$ – the examined subset of attributes; r – resilience level; $\mathbb{S} = (U, A \cup \{d\})$– decision table; $test_C()$ – verifies criterion $\mathbb{C}$

Result: *true – if a given set of attributes* $\breve{R}$ *satisfies selected criteria test*$_{\mathbb{C}}$ *with expected resilience level r, false – otherwise*

```
candidate_subsets ← {X ⊂ R̆ such that |X| = r}
foreach X ∈ candidate_subsets do
    if !test_C(R̆ \ X) then
        return false;
    end
end
return true;
```

the resilient version of $test_{\mathbb{C}}$ function: $r\text{-}test_{\mathbb{C}}$ (Algorithm 3). Algorithm $r\text{-}test_{\mathbb{C}}$ verifies if a given set of attributes $R \subseteq A$ satisfies the resilient criterion r-$\mathbb{C}$ under the condition that implementation of $test_{\mathbb{C}}$ is given.

In Subsection 3.5.1, we present a novel algorithm generating r-$\mathbb{C}$-reducts inspired with a permutation-based technique that is common for RST-based approaches [353, 367]. In Subsection 3.5.2, we follow with discussing the approximation of the permutation-based algorithm for resilient feature selection.

3.5.1 Permutation-based Algorithm

The function $test_{\mathbb{C}}(R)$ verifies if a given set of attributes $R \subseteq A$ satisfies the specified criterion $\mathbb{C}$ in a given decision table $\mathbb{S} = (U, A \cup \{d\})$. Let us assume that we have an implementation of $test_{\mathbb{C}}$ for $\mathbb{C}$. Algorithm 3 shows how to introduce function $r\text{-}test_{\mathbb{C}}(\breve{R})$ that verifies whether the given set $\breve{R}$ satisfies $test_{\mathbb{C}}$ after removal of any r attributes.

Algorithm 4 ($genRed_{\breve{R}}$) generates an r-$\mathbb{C}$-reduct $\breve{R}$ for a given criterion r-$\mathbb{C}$ with the expected resilience level r. The pessimistic computational complexity of $genRed_{\breve{R}}$ with respect to $r\text{-}test_{\mathbb{C}}(\breve{R})$ is $\mathcal{O}(|A|)$, since both loops – *'while'* and *'foreach'* – are iterated at most $|A|$ times. Thus, the computational complexity of the $r\text{-}test_{\mathbb{C}}(\breve{R})$ implementation has a crucial impact on the complexity of $genRed_{\breve{R}}$.

We may notice that for relatively big values of r, e.g., $r = \frac{|A|}{2}$ or $r = \frac{|A|}{3}$, the $r\text{-}test_{\mathbb{C}}(\breve{R})$ may iterate $test_{\mathbb{C}}(\breve{R})$ exponentially many times. However, for any constant r, the algorithm is polynomial. Comparing to the current market standards / defaults with regard to security, resilience and high availability of services and the data, we may notice that data replication levels offered out-of-the-box by database management systems vary near relatively low values (2 to 6) as a reasonable compromise between resilience and storage costs. For example, the default data replication level in most of MapReduce implementations like, e.g., Hadoop[2] is set to 3. Cloud services that offer very high level of durability and availability of stored data usually use between 3 and 6 data replicas, depending on the service pricing

[2] http://hadoop.apache.org

Algorithm 4: $genRed_{\breve{R}}$

Data: r – expected resilience level; $\mathbb{S} = (U, A \cup \{d\})$– decision table; $test_{\mathbb{C}}$– the function that verifies if a given set of attributes $R \subseteq A$ satisfies the specified criteria $\mathbb{C}$

Result: $\breve{R}$ – a subset of attributes that satisfies resilient criterion r-$\mathbb{C}$

```
1  R̆ ← ∅; Permute set A;
2  /* Forward propagation                                   */
3  while r < |R̆| ∧ !r-test_C(R̆) do
4  |  a ← removeNextAttribute(A)
5  |  R̆ ← R̆ ∪ a
6  end
7  /* Backward elimination                                  */
8  foreach a ∈ R̆ do
9  |  if r-test_C(R̆ \ a) then
10 |  |  R̆ ← R̆ \ a
11 |  end
12 end
13 return R̆ ;
```

level[3]. Having that in mind, we may calculate that in the case of $r = 3$ (or $r = 6$ in a very restrictive case) the complexity of r-$test_{\mathbb{C}}(\breve{R})$ relies mostly on the complexity of $test_{\mathbb{C}}(\breve{R})$ and the size of $\breve{R}$. With limited constant r, the function remains polynomial and may be estimated as $\mathcal{O}(|A|^r) \times \mathcal{O}(test_{\mathbb{C}})$ where $\mathcal{O}(test_{\mathbb{C}})$ refers to the complexity of $test_{\mathbb{C}}$. In that case, the complexity of $genRed_{\breve{R}}$ depends on the specifics of $\mathbb{C}$. In the case of, e.g., the classical discernibility criterion $\mathbb{C}^{(Disc,0)}$, the pessimistic complexity of *isReduct* is $\mathcal{O}(|U| \times |A|^2)$.

3.5.2 Approximation Algorithm

Once we have defined the straightforward permutation algorithm, we may elaborate on possible approximations in order to improve the overall feature selection performance. There are plenty of approximation methods that may be adopted to this case like, for example, the DAAR heuristics introduced in [183].

Algorithm 5 ($approximateRed_{\breve{R}}$) follows the idea that $r + 1$ disjoint attribute subsets, which individually satisfy $\mathbb{C}$, constitute a set that satisfies r-$\mathbb{C}$ after being merged together. In the presented pseudo-code we rely on the permutation-based algorithm to construct disjoint $r + 1$ $\mathbb{C}$-reducts. We merge them together to form an r-$\mathbb{C}$-superreduct $\breve{R}$. The set $\breve{R}$ constructed this way, is for sure a r-$\mathbb{C}$-superreduct, since we may remove any r attributes and at least one $\mathbb{C}$-reduct will be untouched. The size of the output is no more than r times bigger than an r-$\mathbb{C}$-reduct could be, which may be still acceptable for highly multidimensional real life data sets.

As a conclusion, in the case of resilient feature selection, the analyst should elaborate on the required level of resilience from the perspective of importance and

[3] https://docs.microsoft.com/en-us/azure/storage/storage-redundancy

Algorithm 5: *approximateRed*$_{\breve{R}}$

Data: r – expected redundancy level; $\mathbb{S} = (U, A \cup \{d\})$– decision table; $genRed_C()$ – the function that generates the set of attributes $R \subseteq A$, which satisfies the specified criteria $\mathbb{C}$

Result: $\breve{R}$ – *a subset of attributes that is a* r-$\mathbb{C}$-*superreduct*

```
1  R̆ ← ∅
2  Permute set A
3  for i = 0; i < r + 1; i++ do
4  |  R ← genRed_C(A);
5  |  A ← A \ R;
6  |  R̆ ← R̆ ∪ R;
7  |  if A = ∅ then
8  |  |  break; /* If we tested all attributes                    */
9  |  end
10 end
11 return R̆ ;
```

sensitivity of the problem. Nevertheless, one should have in mind the impact of resilience level on the performance of feature selection and should adjust it with respect to the aforementioned factors. On the other hand, minimal subsets of attributes are not always desired. In some situations, it is worth combining the groups of approximate reducts in order to improve performance of prediction models [138]. This shows that properly managed redundancy in selected attribute sets may not only increase the resilience of the solution but also may have a positive impact on the quality of trained models.

3.5.3 Algorithm Working Example

In order to provide a better understanding of the presented algorithms and to verify the quality of the proposed approach, let us experiment with the size of r-$\mathbb{C}$-reducts acquired by Algorithm 4 for the data in Table 2.2 and conditional entropy H, namely the criterion function $\mathbb{C}^{(H,\ 0.2)}$. The resilient version of criterion r-$\mathbb{C}^{(H,\ 0.2)}$ may be constructed for a given resilience level $r = 1$ as described in equation (3.9).

Below, we present the step-by-step description of a single execution of Algorithm 4. Afterwards, we present a summary of 10 independent executions and the statistical analysis of the expected size of 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts for the data in Table 2.2.

During the first execution of the experiment, we sorted the set A (line 2 in Algorithm 4) with the permutation σ. Within the *'while'* loop (lines 3-6), the algorithm iterated over A according to $\sigma_1 : a_6, a_3, a_1, a_4, a_5, a_2$. After consecutive draws of attributes, we evaluated whether the condition 1-$\mathbb{C}^{(H,\ 0.2)}$ was met. Below we enumerate each iteration of the *'while'* loop during the first execution of the experiment:

- 1st iteration
  1. *removeNextAttribute*(A) returns a_6, hence $\breve{R} = \{a_6\}$

Table 3.4: Summary of the experiments for Algorithm 4.

i	σ_i	1-$\mathbb{C}^{(H,\ 0.2)}$-reduct	size
1	$\sigma_1 : a_6, a_3, a_1, a_4, a_5, a_2$	$\{a_1, a_6\}$	2
2	$\sigma_2 : a_2, a_3, a_4, a_1, a_5, a_6$	$\{a_2, a_3, a_4\}$	3
3	$\sigma_3 : a_5, a_3, a_2, a_1, a_4, a_6$	$\{a_2, a_3, a_5\}$	3
4	$\sigma_4 : a_1, a_3, a_2, a_4, a_5, a_6$	$\{a_1, a_2\}$	2
5	$\sigma_5 : a_1, a_5, a_3, a_2, a_4, a_6$	$\{a_1, a_3, a_5\}$	3
6	$\sigma_6 : a_6, a_4, a_2, a_5, a_3, a_1$	$\{a_2, a_6\}$	2
7	$\sigma_7 : a_2, a_5, a_1, a_6, a_3, a_4$	$\{a_1, a_2\}$	2
8	$\sigma_8 : a_4, a_3, a_2, a_1, a_5, a_6$	$\{a_2, a_3, a_4\}$	3
9	$\sigma_9 : a_6, a_2, a_1, a_5, a_4, a_3$	$\{a_1, a_6\}$	2
10	$\sigma_{10} : a_5, a_3, a_6, a_2, a_4, a_1$	$\{a_3, a_5, a_6\}$	3

2. 1-$test_{\mathbb{C}}(\{a_6\}) = 0$, because $\breve{R} \setminus \{a_6\} = \emptyset$

- 2nd iteration

1. $removeNextAttribute(A)$ returns a_3, hence $\breve{R} = \{a_6, a_3\}$
2. 1-$test_{\mathbb{C}}(\{a_6, a_3\}) = 0$ because for the subset $\{a_3\}$, $H(d|\{a_3\}) = 0.74$, hence $H(d|\{a_3\}) > -\log_2(1 - 0.2)$

- 3rd iteration

1. $removeNextAttribute(A)$ returns a_1, hence $\breve{R} = \{a_6, a_3, a_1\}$
2. 1-$test_{\mathbb{C}}(\{a_6, a_3, a_1\}) = 1$ because for all subsets $\breve{R}' \in \{\{a_1, a_3\}, \{a_1, a_6\}, \{a_3, a_6\}\}$, we have $H(d|\breve{R}') \leq -\log_2(1 - 0.2)$

In the first execution, in the *'foreach'* loop (lines 8–12), we iterated over $\breve{R} = \{a_6, a_3, a_1\}$ trying to eliminate superfluous attributes. The attribute a_6 could not be removed because 1-$test_{\mathbb{C}}(\{a_1, a_3\}) = 0$. The attribute a_3 was removed with no impact on 1-$\mathbb{C}^{(H,\ 0.2)}$ because $H(d|\{a_1\}) \leq -\log_2(1 - 0.2)$ and $H(d|\{a_6\}) \leq -\log_2(1 - 0.2)$, thus 1-$test_{\mathbb{C}}(\{a_1, a_6\}) = 1$. The last attribute a_1 could not be removed. This way, we reached 1-$\mathbb{C}^{(H,\ 0.2)}$-reduct $\breve{R} = \{a_1, a_6\}$ of size $|\breve{R}| = 2$ – which is the minimal possible.

To provide higher reliability, the experiment was repeated 10 times for 10 randomly chosen permutations. Table 3.4 summarizes each iteration *'i'*, including permutations $\sigma_1, \ldots, \sigma_{10}$ and the derived 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts $\breve{R}_1, \ldots, \breve{R}_{10}$ with their size.

Lastly, let us elaborate on the expected size of 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts that may be generated with Algorithm 4. The minimal size of 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts is 2, whereas the maximal size of 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts in the analyzed data is 3, because for each two-element set $R \subset A, H(d|R) \leq -\log_2(1 - 0.2)$. There are 6! possible permutations σ of A. Let us estimate the number of permutations that would result in 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts of size 2. Such permutations should have two of attributes $\{a_1, a_2, a_6\}$ within the first 3 attributes. There are $\binom{3}{2} * \binom{3}{1} = 9$ three-element combinations that contain two attributes of $\{a_1, a_2, a_6\}$ and one that contains all.

Each of them may be permuted in 3! ways and the rest of σ may be arranged in 3! ways. So, $\frac{10*3!*3!}{6!} = \frac{1}{2}$. The rest half of permutations would result in 1-$\mathbb{C}^{(H,\ 0.2)}$-reducts of size 3. Thus, the expected size of 1-$\mathbb{C}^{(H,\ 0.2)}$-reduct generated by Algorithm 4 for the data in Table 2.2 is equal to $\frac{1}{2} * 2 + \frac{1}{2} * 3 = 2.5$.

Chapter 4

Technical Aspects of Interactive Feature Engineering

Several useful techniques have already been applied for decision making in the domains of data warehousing (DW), business intelligence (BI) and machine learning (ML) [387]. However, before one may apply machine learning algorithms on the collected data, several steps need to be performed in the first place [84, 424]. Among them, feature extraction and selection are recognized as the most challenging, time-consuming and computationally cost-full [74]. Optimistically, the similarity of the nature of sensory and machine generated data provides an opportunity to construct generic, reusable mechanisms for interactive data processing, exploration and analysis. In this chapter, we introduce a new approach for learning forecasting models over large multi-sensor data sets, including the steps of sliding window-based feature extraction and ensemble-based feature selection.

4.1 Sliding Window-based Feature Engineering

In this section, we outline our approach to feature extraction, aimed at processing data obtained from sensors that monitor certain changes in the environment and provide outputs in a form of time series. Individual readings may take different forms according to the application domain [267,300]. Values may express continuous phenomena, such as pressure, humidity, or the level of methane concentration in a longwall of a coal mine [197]. They can also express a discrete state of the environment, such as an on/off state of a device or vehicle movement direction. To acquire knowledge about environment state and its changes, it is common to set up a collection of sensors, potentially of different types. Therefore, the gathered data elements can be complex on various levels and sometimes their interpretation is possible only in a context of additional knowledge obtained from domain experts. This, in turn, requires appropriate mechanisms for human-system interaction. No less important is the ability to properly parallelize data processing in order to deal with various challenges related to Big Data [16, 91].

4.1.1 Prerequisites and Data Preprocessing

Information systems - especially in the field related to reporting, business intelligence, machine learning, and decision support - integrate data from many other systems and sensors [356]. Having data from various sources in a single place provides a valuable opportunity for pattern discovery. However, those systems and sensors are usually provided by many different corporations, and are developed using various technologies with different data formats. Information provided by each can be textual, categorical, numeric, etc. All may also refer to different data dictionaries and taxonomies. Therefore, prior to feature extraction, let us outline some typical challenges related to data integrity and quality as well as basic steps of data preprocessing which becomes particularly important when real streams of sensor readings are involved.

In the first place, the analysis of a big variety of data representations requires some kind of unification protocol. With that respect, let us refer to so called *sensor card* – an information template created on the basis of investigation of a large variety of sensors that can be applied whenever heterogeneous data sources need to be systematized, so that could be integrated. An example of a sensor card - a common interface that allows to describe in a consistent way a variety of types of sensors and devices that are used in the domain of underground mining - is presented in [356]. This way, the sensor readings from different data sources may not only be integrated but also further enhanced with some specific features highly depended on the analyzed problem, like: the organization of the shift work, shift schedule and plan, information about bank holidays, recent local events, etc.

The industrial monitoring systems usually produce multidimensional streams of sensor readings for which performing standard preprocessing steps such as data integration, data cleaning, feature extraction and selection, etc. is quite challenging. Measurements recorded by sensor devices tend to be noisy. Because of faults and errors that may have place in real environments, it is also difficult to maintain decision models that should be used in an on-line fashion. Thus, the goal of data preparation and cleaning is to translate data to a form acceptable by the forecasting model construction methods. This phase is focused on the preparation of the training sets for further analysis and, once the models are ready, becomes responsible for feeding them with new inputs. Let us outline some typical issues connected to the data acquisition in real life environments:

1. *Unsynchronized readings*: Reading frequencies differ for different sensors.
2. *Missed readings*: Sensors may stop delivering in a given time interval.
3. *Outlier readings*: Sensor readings are frequently imprecise or unreliable.
4. *Rare readings*: Usually, the most critical events occur in data very sparsely.

The first task is to adjust sensor readings that are collected at different frequencies. Also, some systems collect a new reading only in the case of a sufficiently significant change of the measured value. The main objectives of subsequent tasks are imputation of missing values, and outliers detection [338, 362].

he imputation of missing data in time series is a particularly difficult task [27], and many general techniques are not able to satisfactory deal with this case. And the

subcase of multivariate time series stays at the core of the most challenging tasks, as observed in

This are particularly complex tasks for multivariate time series and spatio-temporal data environments [15,46]. With this respect, one can, e.g., create a logical expression that defines value replacements (for instance, to replace values < 1 with "low state"), use a default value, follow the last valid reading, take an average of the neighboring readings, or apply linear regression based on the preceding values. Imputation techniques are often based on univariate series analysis or sampling from original data distribution [338, 362]. There are also approaches based on auto-regressive models constructed, e.g., by combining an expectation-minimization algorithm with the prediction error minimization method, or based on Bayesian models [24]. The maximum allowed number of consecutive missing values that can be imputed should be set up too. On the other hand, the missing value imputation (or outlier replacement) is not always a requirement. It may depend on a context of a given sensor and knowledge about its operation conditions, or simply on further preprocessing steps which can deal with incomplete data on the fly.

In the proposed approach, a sliding time window method allows us to overcome, or at least to minimize the impact of, some of the above-mentioned issues related to time series data. A viable approach to deal with missing or unreliable attributes - the resilient feature selection with r-$\mathbb{C}$-reducts – was proposed in Chapter 3. Later in Section 4.2.2, we discuss another approach to overcome problem with missing data basing on feature selection over granular attribute space due to attribute interchangeability. Whereas, in Section 4.3, we propose ensemble-based feature selection method which is also a practical approach to introduce a certain redundancy of information.

A special case of data preprocessing corresponds to the creation of a dependent variable. This is a crucial aspect for any supervised learning approach. For this purpose, we may use a dedicated operator which allows us to define a dependent variable as the maximal value measured for a given sensor within a specified time interval (e.g., three to six minutes into the future). This can be considered as an example of a broader window-based methodology described below. Such a style of specifying a dependent variable may also decrease sparsity of its critical values. This is because a single high measurement influences a score of the whole time interval. Given the above steps of data preprocessing, we are now ready to go to the topic of feature extraction.

4.1.2 Sliding Windows

A sliding window travels through the time series from the beginning to the end and replaces a sequence of raw sensor readings with some of its derived statistics, accordingly to the predefined aggregation functions (cf. Table 4.1). The range of aggregation can be chosen by the users by means of, e.g., a time unit that defines windows containing sensor readings to be grouped together. For each outcome of aggregation, we can calculate a weight corresponding to the quality of the original data that is, e.g., inversely proportional to a number of missing values or outliers involved. This approach allows us to reduce the number of missing values in data,

Table 4.1: Examples of features that represent each time window.

Feature type	Description
basic meta information	e.g., a data source identifier, ID or a name of the sensor, a total number of readings "n", a number of valid readings "nValid", etc.
quality assurance and reliability of a given windows	e.g., a ratio of correct readings in the window $= \frac{nValid}{n}$, or just a number of identified outliers or missing values
time-range of a window and time related data	e.g., year, month, day of month, day of week, hour, time-range, etc.
basic summary of all readings in a given time window	statistics: mean, min, max, stdDev, median and pecentiles: $5th$, $10th$, $25th$, $75th$, $90th$, and $95th$, etc.
transformations and measures of values in time window	e.g., selected Fourier transform coefficients, skewness, Kurtosis measure, etc.
summaries of consecutive sub-windows	the same statistics as above, computed for sub-windows of a given window
trends related to recent readings in a time window	differences between the last reading and min/max values, differences between last/first readings in a window
statistics for differentiated values in a time window	mean, median, min, max, stdDev and percentiles of differences between two consecutive readings
measures derived from summaries of a time window	differences between min and max, mean and median, max and percentiles, etc.
measures derived from summaries of sub-windows	differences between quantities of mean, median, min, and max values representing consecutive sub-windows
indicators of extreme readings	position in a time window of a reading with min/max value, position of a min/max value in the latest sub-window
indicators of extreme transformed readings	position of a maximum difference between consecutive readings, etc.
a set of values that express the trend between statistics in consecutive sliding windows	inter window data, e.g., a difference between *min*, *max*, *mean* or between Xth percentiles, where: $X \in \{5, 10, 25, 50, 75, 90, 95, etc.\}$

and to introduce weights that can be utilized further by analytical methods. It is also worth mentioning that such aggregation operations can work on multiple sensor readings with unsynchronized frequencies [356].

Processing of a single time series requires two parameters: *length* and *offset*. The first of them defines the size of a sliding time window, e.g., the number of readings to be involved, or a corresponding time interval. The alignment of processed time windows is controlled by the second parameter. It defines a degree to which two consecutive windows overlap each other. Let us here recall Figure 2.1 – presented in a preliminary Section 2.1 – which highlights four examples of sliding window set-ups. The first example, marked in red, shows the situation when the length of a sliding window is equal to the offset. The green and blue examples show the consecutive positions of a sliding window when the offset is equal to $\frac{1}{2}$ and $\frac{1}{3}$ of the length, respectively. The system is also capable to express the situation when the offset is

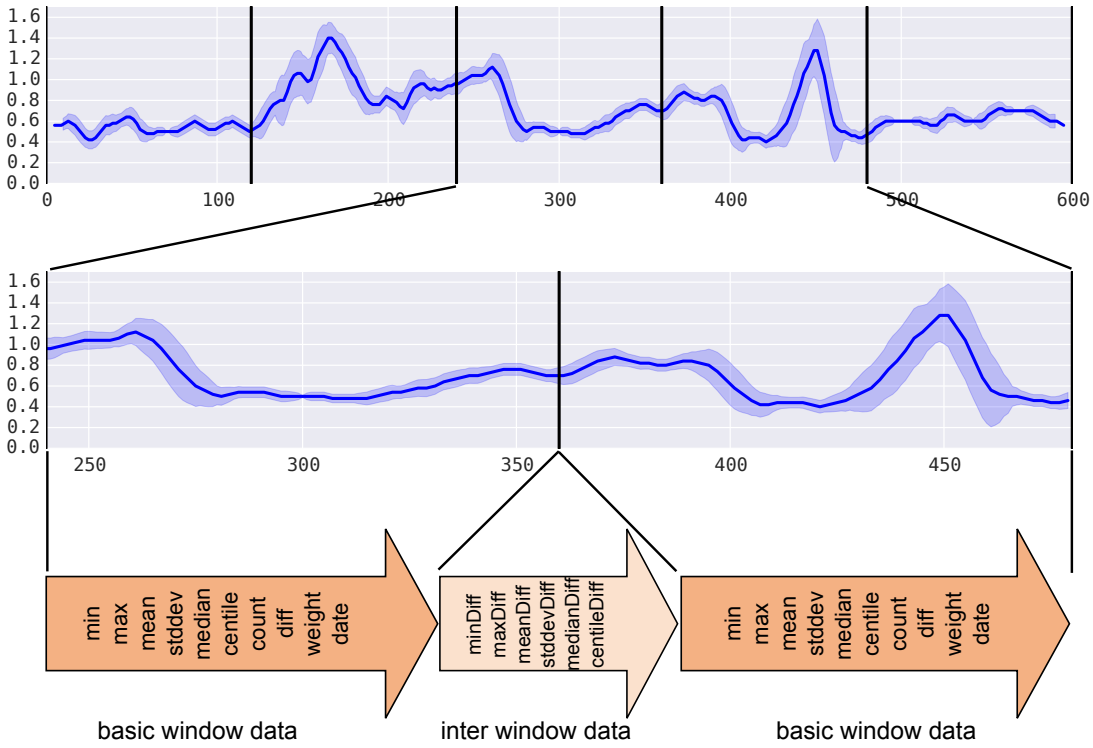


Figure 4.1: Overview of the proposed sliding window approach. The topmost time series is split into five non-overlapping basic windows of equal length. Two of them are zoomed in the middle of the above diagram. The statistics are computed for each window separately. Inter window features express the dynamics of changes of basic window statistics.

greater than the length – the example marked in cyan.

A collection of sensors that perform readings produces a corresponding collection of time series data. A single time series is an ordered sequence of readings associated with the timestamp at which it was collected. A collection of aggregated values created from a time series may be organized arbitrarily at a higher level. For instance, if a time window covers one minute, then we may be interested in five consecutive windows that cover five minutes of data in order to analyze various trends over derived statistics within that period. It is quite different than aggregations over a single five-minutes time window [138, 143]. Going further, sensors may correspond to many different time series processed independently. To obtain a complete description of the environment at a given time point, time series collected from different sensors should be combined together to form a larger set of aggregated values [144].

During the process of moving a time window through a time series, each of its fixed positions defines so-called *basic window* (cf. Figure 4.1). For such a basic slice of data, a predefined aggregation functions are applied. Each aggregation function can be seen as a new window feature. This step may be adjusted to a specific data domain by supplying different aggregation implementations [144, 179, 356]. The proposed set of features which are calculated to represent the data in a basic window are presented in Table 4.1. If we consider more than one consecutive basic window to represent the environment state at a given time point, then we can extract so-called *inter window features* expressing trends and changes between pairs of basic windows.

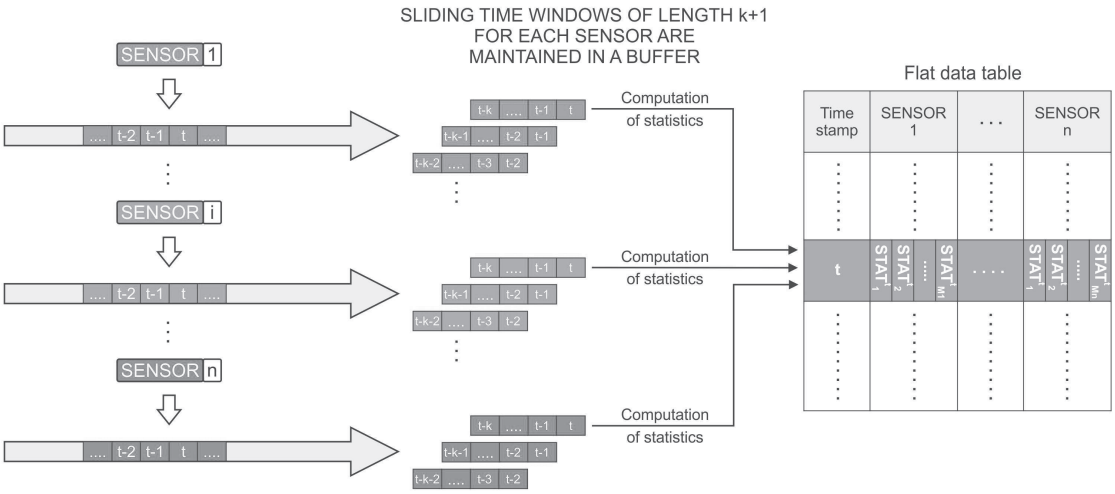


Figure 4.2: Window-based features calculated for a portion of time series data.

Some examples of inter window features that are calculated to represent the changes between two consecutive basic windows are presented in Table 4.1. A schematic view of extracted basic window features, inter window differential statistics as well as their relation to the processed time series is shown in Figure 5.

Yet another very specific approach is the extension of the sliding window construction process by adding some more static attributes reflecting assessments obtained from domain experts [179]. This brings the opportunity to compare the prediction quality of models trained using derived features with the expert-based assessment as well as makes it possible to use features derived from experts in ML models training. Furthermore, the expert assessment helps to address the so-called *cold start* problem, when a decision support system is installed, e.g., in a new location and it does not yet have a sufficient amount of data to fit into the new environment [364].

The proposed framework is capable of operating on multidimensional time series derived from a number of sensors. The default method of processing multiple series is hierarchical, i.e., each time series is processed independently and then the results are combined according to specified settings. Afterwards, depending on configuration, features corresponding to basic windows and inter window features derived for selected sensors create so-called composite windows which represent the overall state for a given time point.

This way we provide a comprehensive data preprocessing and feature extraction framework that can be used for constructing informative and robust representation of multidimensional time series data, as visible in Figure 4.2. The overall mechanism of computing time-window-based representations can be treated as a universal approach. It is worth noticing that due to a diversity of extracted features and a high number of considered sensors this representation of data may be highly dimensional. Hence, it may require feature selection before forecasting model construction techniques could be applied.

4.2 Feature Space Granulation in Feature Selection

The process of feature selection aims at exploring the given attribute space A (or A^*) and extracting a relatively small subset $R \subseteq A$ of attributes that, on the one hand, are the most relevant and, on the other hand, are sufficient to solve the investigated problem. Such selection/extraction process is often conducted by applying statistical tests in order to determine, which attributes contribute to the constructed decision model [149]. Although the standard feature selection algorithms are not configured for attributes that are structured or bound by relationships, the knowledge about attribute granulation can have an important impact on the final subset composition.

4.2.1 Feature Space Granulation

The attribute granules can take various forms. It is possible to group or cluster features on the basis of their relationship, and it can be done in a parameterized manner. For example, we can produce various versions of granulations depending on the choice of cutoff value after the original attributes are hierarchically clustered [139]. In this context, it is important to have the means of assessment of the resulting granules, similar to those developed for standard data clustering. By making the feature selection process aware of the underlying granular structure of attribute space one can make better use of the knowledge contained therein. This in turn may lead to selecting the sets of features that are not only optimal from the perspective of some mathematical criteria but are also more useful for interpreting knowledge hidden in the data.

Let us now present two specific examples of the granulation based on the attribute interchangeability. The first approach is centered around the notion of *explicit interchangeability* of features in attribute subsets that are small in size but sufficient to model the target decision classes/labels. In the theory of rough sets, such attribute subsets are usually referred as decision reducts (Definition 1). Intuitively, if two attributes rarely belong to the same subset but they both often appear together with similar groups of other attributes, they may be considered interchangeable. In the opposite situation, when two attributes often belong to the same subset or appear in a company of completely different features, it seems reasonable to assume that they convey different information and thus are not similar. More formally, this type of attribute interchangeability can be measured using a co-occurrence frequency matrix F, whose entry in *i-th* row and *j-th* column equals $f_{i,j}$:

$$f_{i,j} = \frac{|\{k : a_i \in R_k \wedge a_j \in R_k\}|}{|\{k : a_i \in R_k\}|} \tag{4.1}$$

where a_i, a_j are attributes, $i \neq j$ and R_k is the *k-th* pre-computed attribute subset (reduct). All values at diagonal of F are set to 0. The final values of attribute interchangeability can be computed as a difference between the similarity of corresponding feature sets and the frequency, with which the given features co-occur, e.g.:

$$I(a_i, a_j) = cosine(f_{i,\cdot}, f_{j,\cdot}) - f_{i,j} \tag{4.2}$$

In this formula, $f_{i,\cdot}$ and $f_{j,\cdot}$ are vectors of values from *i-th* and *j-th* rows of F, respectively. Such an approach was successfully applied in [180]. Figure 4.4 depicts a heat map of a distance matrix that was used for identifying key risk factors for firefighters during fire&rescue actions. Analogous heat maps could be computed using, e.g., ensembles of possibly diverse *approximate* decision reducts that preserve information about the decision classes only to some extent and, thus, they can utilize different groups of attributes to concentrate on different aspects of approximate data dependencies [353].

A slightly different approach is centered around the attribute similarity function sim^{Disc}, which refers to the discernibility relation (2.4) and its numeric representation - measure $Disc$ (2.5). For a given decision system $\mathbb{S} = (U, A \cup \{d\})$, we may define $sim^{Disc} : A \times A \to \mathbb{R}$ as follows:

$$sim^{Disc}(a, a') = \frac{|\{(u, u') : d(u) \neq d(u') \wedge a(u) \neq a(u') \wedge a'(u) \neq a'(u')\}|}{|\{(u, u') : d(u) \neq d(u') \wedge (a(u) \neq a(u') \vee a'(u) \neq a'(u'))\}|} \tag{4.3}$$

where $(u, u') \in U \times U$ and $a, a' \in A$. So defined attribute similarity measure expresses a ratio between a number of pairs of objects from different decision classes that are discerned by *exactly one* attribute from the considered pair, to a number of such objects discerned by *at least one* of the compared attributes.

We may extend the definition of the above attribute similarity measure sim^{Disc} (4.3), so that it operates on subsets of attributes instead of individual attributes. For a decision system $\mathbb{S} = (U, A \cup \{d\})$, the attribute subsets similarity function $Sim^{Disc} : \mathcal{P}(A) \times \mathcal{P}(A) \to \mathbb{R}$ is defined as:

$$Sim^{Disc}(R, R') = \frac{|\{(u, u') : \exists_{a \in R, a' \in R'}\ d(u) \neq d(u') \wedge a(u) \neq a(u') \wedge a'(u) \neq a'(u')\}|}{|\{(u, u') : \exists_{a'' \in R \cup R'}\ \ d(u) \neq d(u') \wedge a''(u) \neq a''(u')\}|} \tag{4.4}$$

4.2.2 Feature Selection Algorithms with Attribute Granules

In this section, we examine to what extent feature granulation can guide the process of choosing the most appropriate collections of attributes. We argue that it should influence the order, in which we investigate attributes. We discuss the meaning of similarity, proximity and functionality while considering the granules of physically existing, or potentially derivable attributes in the feature extraction process. We also propose several approaches to utilize granulation structures defined over the feature spaces in feature selection algorithms. In particular, we consider the algorithms developed within the theory of rough sets, aimed at finding irreducible subsets of attributes that are sufficient to distinguish between the cases belonging to different target decision classes.

For the purpose of further discussion, let us concentrate on the approach to conducting feature selection proposed in Section 3.1. Certainly, we do not claim that all possible methods follow the scheme below. Nevertheless, it is sufficiently general to explain the benefits of working with attribute granules. For a given decision system $\mathbb{S} = (U, A \cup \{d\})$ and input set of attributes A, let us consider a criterion function $\mathbb{C} : \mathcal{P}(A) \to \{0, 1\}$ (cf. Definition 5) that indicate which subsets of attributes are already rich enough to serve as the outcomes of the selection process. In practice, $\mathbb{C}$ may correspond to a collection of criteria reflecting different requirements. Additionally, let us consider an arbitrary heuristic quality function $Q : \mathcal{P}(A) \to \Theta$ (cf. Definition 7) that can be utilized iteratively to add the most "promising" elements to the constructed feature subset. Let us note that Q can combine various aspects of relationships between the selected attributes and a target variable [80, 117, 174, 294]. Let us also mention that the last item of the following procedure has strong roots in the theory of rough sets, where there is a particular focus on the simplification of decision models [75, 370, 412].

1. While the criterion $\mathbb{C}(R)$ is not met by the selected feature subset R continue the following:
   (a) Select candidate subsets of features $B_1, .., B_k$ to be added to R
   (b) Evaluate $B_1, .., B_k$ with the desired attribute subset quality measure Q
   (c) If the best B_x contributes to R, then $R \leftarrow R \cup B_x$
   (d) Verify if the criterion $\mathbb{C}(R)$ is met
2. Eliminate superfluous attributes from R

Algorithm 6 reflects our generic idea of embedding the additional knowledge about attribute granulation into the above-described feature selection process. In each iteration of the main loop, in order to limit the attribute space A, the subset of granules $\{G_1, .., G_m\} \subseteq \mathbb{G}$ is selected with respect to the granulation preferences expressed by, e.g., a permutation $\sigma_{\mathbb{G}} : \{G_{\sigma(1)}, G_{\sigma(2)}, ..\}$ (which means that the granule $G_{\sigma(1)}$ is most preferred to draw attributes from). By limiting the search space using the additional knowledge about attribute granulation, we may quickly generate a set of candidates $\{B_1, .., B_k\}$. After the evaluation of candidates with the correlation, Gini index, or other implementation of the function Q, the feature subset R may be extended if only the selected B contributes to R. The loop continues until a "good enough" R is collected or all combinations/candidates are explored. Finally, we conduct a backward elimination of superfluous attributes.

The presented framework does not enforce any particular interpretation of the information granules and, thus, different implementations may vary in a way of their utilization. In some cases, it may be preferred to select features that belong to only one, specific granule. For example, the analysis of coal mine sensor readings may be oriented on the one, particular mine shaft [179]. In that case, the analyst could generate granules on the basis of a sensor location and introduce a constraint that the finally selected attributes should/must belong to the particular ones. In other applications, it may be convenient to generate an attribute subset that contains

Algorithm 6: General framework for granular feature selection

Data: $\mathbb{G}$ – set of granules, A – attribute space,
$\mathbb{C}$– criterion function, $\sigma_{\mathbb{G}}$ – granule preferences
Result: R – selected attribute subset

```
/* Initialization                                          */
R ← ∅
while R does not satisfy ℂ(R) do
    B ← ∅
    Select granules {G_1, .., G_m} ⊆ 𝔾 with respect to σ_𝔾
    Limit attribute space A_G ← A ∩ ⋃_{1≤i≤m} G_i
    Generate candidates B_1, .., B_k ⊆ A_G
    Evaluate candidates {B_1, .., B_k}
    B ← selectBestCandidate({B_1, .., B_k}, ...)
    if B contributes to R then
        R ← R ∪ B
    end
end
R ← eliminateSuperfluousAttributes(R)
return R;
```

attributes from multiple granules in order to provide higher robustness [3]. Regardless of the way that we use the attribute granulation, the general framework is still the same.

Attribute granulation may also influence a feature selection process with respect to the expected robustness and resilience of decision models. In real-life applications, we may observe various anomalies in explored data sets, which cause a model over-fitting. Some researchers emphasize the role of appropriate granulation of attributes during feature engineering in achieving higher stability of the created models. With that respect, we may refer to several techniques using, e.g., clustering or histograms [422]. During the decision model construction, there are also some non-functional factors that could impact the continuity of analysis like, e.g., temporal or permanent unavailability of some sources during on-line data collection [137]. From this perspective, it is advisable to use diverse feature subsets and ensemble methods, whereby each of separate decision models is based on a few attributes but, overall, many attributes are involved [182]. Thus, it is important to combine the feature selection approaches relying on the attribute granulation with some feature subset diversification methods.

In this context, the objective is to achieve more robust and resilient results due to, e.g., exploitation of attributes extracted from diverse sources. In particular, the method outlined by Algorithm 6 could be used to compose an attribute subset R as a collection of features from diverse granules. In this case, the attribute reduction algorithm should aim at achieving feature subsets of minimal cardinality $|R|$ and also ensure the diversity of granules by, e.g., maximization of $|\{G \in \mathbb{G} : R \cap G \neq \emptyset\}|$. Accordingly, a specialized configuration of the main loop in the presented framework can take into account, both, the so-far-selected features and the granules that are used

Algorithm 7: Full-granule-oriented version of Algorithm 6

Data: $\mathbb{G}$ – set of granules, A – attribute space,
Q– quality function, $\mathbb{C}$– criterion function
Result: R – selected attribute subset

```
/* Initialization                                          */
R ← ∅
while R does not satisfy ℂ(R) = 1 do
    Select granules {G_1, .., G_m} ⊆ 𝔾
    Evaluate granules {G_1, .., G_m} and pick the best G_x
    if G_x contributes to R then
        R ← R ∪ G_x
    end
end
R ← eliminateSuperfluousGranules(R)
return R;
```

less often, i.e., granules G_i that minimize the quantity of $|G_i \cap R|$. The presented approach may be considered as a practical solution to the problem of resilient feature selection introduced in Chapter 3.

The feature selection methods should be also able to operate on the whole granules or their subsets instead of individual attributes. To some extent, it corresponds to the idea of so-called *decision systems with constraints* – the enriched data representation proposed in [277]. The goal of this approach is not only to record the presence of granules (called constraints) but also to make it possible to apply various computational methods that make use of them. Let us consider Algorithm 7, where the overall scheme is aligned with Algorithm 6, though one can notice some simplifications like selecting particular granules $G_1, .., G_m$ as the candidate subsets B_i. Similarly, the backward elimination concerns removal of the whole granules instead of individual attributes. In such approaches, as it was observed also by other researchers, the properties of selected attribute subsets can depend a lot on coarsening or refining granules [193]. Therefore, there is a need for a framework allowing the domain experts and algorithm designers to assess the results of feature selection/granulation processes from different perspectives.

As we could see above, Algorithm 6 can be treated as a general umbrella for various approaches aiming at utilization of the attribute space granulation for the purpose of enhancing the feature selection process. Surely, there are still several details to be discussed. First, it is useful to look at different strategies of validating whether a given attribute sufficiently *contributes* to the result R [183]. Second, it is interesting to compare the proposed framework with methods based on attribute orderings. The main idea behind this class of methods is to iterate along diversified permutations σ_A over A. Such permutations can be induced partially with respect to some heuristic function Q, or they can be generated fully randomly [367]. In the latter case, the procedure is repeated a number of times and the best of the obtained attribute subsets (or a bigger ensemble of subsets) is eventually selected.

Figure 4.3 shows how we can use the knowledge about granules to influence

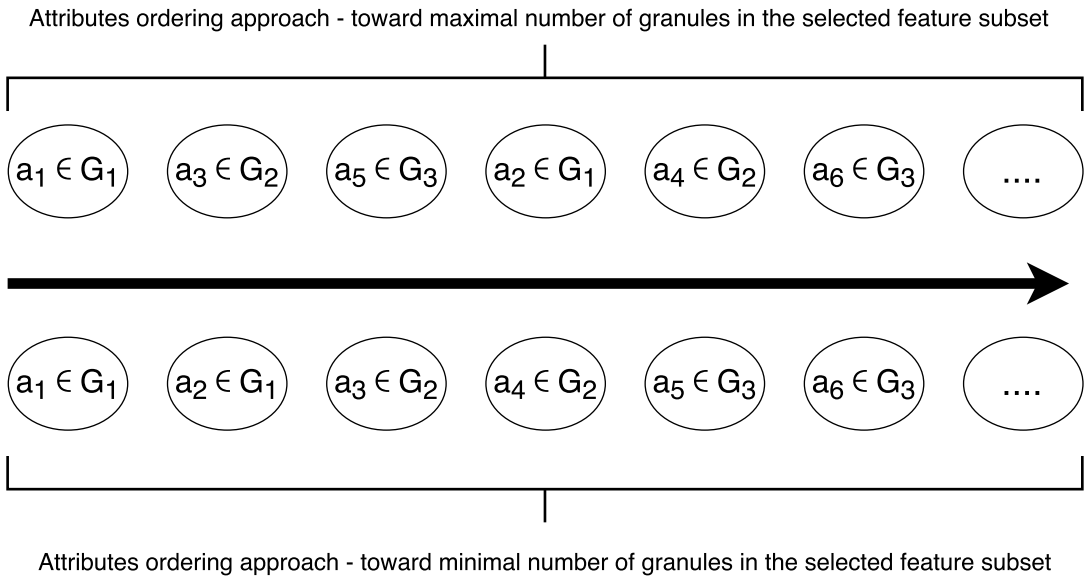


Figure 4.3: A diagram with two significantly different attribute ordering strategies that take into account granulation of attributes.

permutations of attributes, e.g., by arranging the elements of the same granule within consecutive subsequences, or mixing them together as much as possible (by following a "preference" permutation $\sigma_{\mathbb{G}} : \{G_{\sigma(1)}, G_{\sigma(2)}, ..\}$). It is important to note that such two semi-randomized strategies are in a correspondence to the ideas of operating with regular granules (Algorithm 7) and maximally diverse attribute subsets, respectively. This shows that the attribute granulation is easily applicable to the ordering-based feature selection algorithms, without a necessity to modify their code. On the one hand, the described scenarios of "granular ordering" are conceptually aligned with Algorithm 6. On the other hand, the phase of selecting granules/candidates can be performed implicitly at a level of generation of attribute permutations.

While the "case-oriented" granulation is a way to cope with ever-growing amounts of the data, the "attribute-oriented" granulation may turn out to be useful for high-dimensional data problems, whereby the amounts of possible features become difficult to handle. This is visible at the stage of feature selection that is aimed at deriving compact sets of attributes that can be an appropriate input to construct the final decision models. Computational complexity of typical feature selection algorithms depends heavily on the number of potentially useful and derivable features, therefore, any ideas how to reasonably introduce granulation into the feature space are essential for the Big Data.

4.2.3 BigData Aspects of Attribute Granulation

In this section, we discuss how the concept of granulation can be made useful in selecting and engineering features on big and possibly complex data sets. We show how to utilize the intrinsic properties of the data and underlying problem as well as background/domain knowledge for the purpose of building granular representation of attributes. All the provided tools and examples are devised to work with data

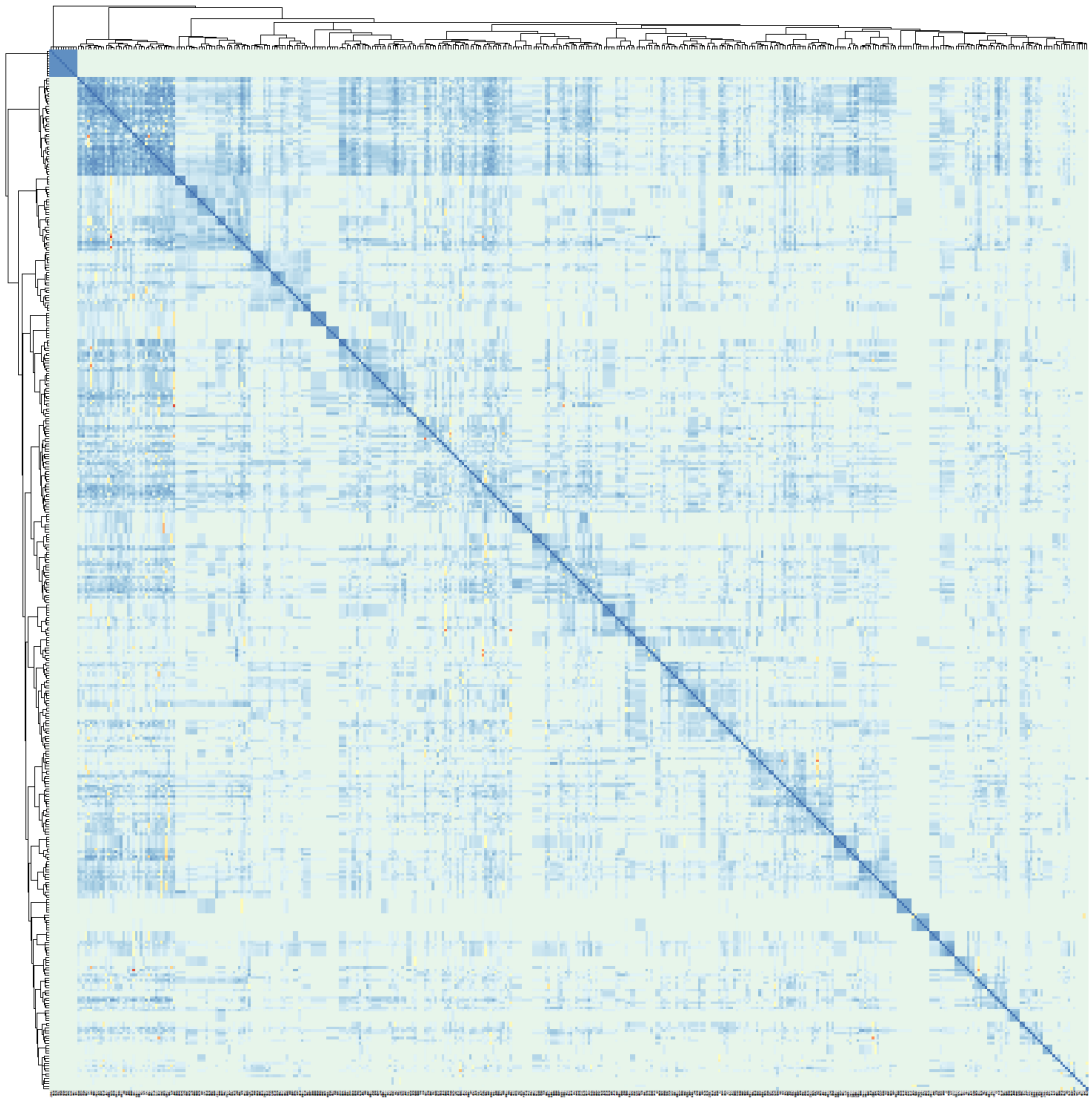


Figure 4.4: A heat map expressing interchangeability of risk factors (represented as attributes) taken from the AAIA'14 Data Mining Competition [180]. Granules of attributes are arranged along the diagonal of the matrix [139].

sets that are very large in terms of the number of objects, as well as the number and complexity of features. Thus, we address at least some of the challenges posed by the Big Data paradigm.

Big Data is often characterized by presence of "Five Vs" – *Volume*, *Variety*, *Velocity*, *Variability*, and *Veracity* – reflecting in the enormous complexity that directly impacts the aforementioned data processing [16,91]. They make it a challenge to represent the task at hand in a way that is at the same time computationally useful and comprehensible for the user. Domain experts expect intuitive data representation, whereas machine generated data collected from, e.g., large sensor networks may have all of the required technical properties but may be hardly understandable and detached from the real-life phenomenon that it is meant to record. The variety of incompatible data formats and non-aligned data structures spanning across photographs, sensor data, tweets, text documents, encrypted packets, etc., can

Algorithm 8: Granular feature selection with iterative MapReduce. In each of ℓ-phases the following program is executed.

Map(Key: $a \in A$**, Value:** G_a, V_a**)** **:**

Given $\mathbb{R} = \{R_1, .., R_n\}$
foreach $R_i \in \mathbb{R}$ **do**
  if *a is relevant to* R_i **then**
    $R_i \leftarrow R_i \cup \{a\}$
    emit(sortAttributes$(R_i), \sigma^i_{\mathbb{G}}$, score)
  end
end

Reduce(Key:$R_i, \sigma^i_{\mathbb{G}}$**, Value:** $\{score, score, ..\}$**):**

emit($R_i, \sigma^i_{\mathbb{G}}$, score)

make it hard to perform data analytics. The possible reduction and transformation of the data set provided by "classical" object-wise granulation mostly addresses the *Volume*, with some additional, lesser impact on *Velocity* and *Variability*. With granular feature selection and construction it is possible to take care of the other "Vs", in particular *Variability* and *Veracity*.

The high velocity and volume of still-incoming records are often a curse of storage systems and machine learning algorithms. Furthermore, raw records are often insufficient for the purpose of predictive analysis and the process of feature engineering is commonly employed to construct more relevant attributes [9]. The massively parallel feature engineering methods may be efficiently performed via the MapReduce programming model what, in turn, may multiply the initial number of explored attributes [138]. Still, the question remains how to choose which attributes should be evaluated. As suggested in [393], the actual feature selection process can be performed at a level of general labels of some attribute granules, whereby specific elements of those granules are not materialized prior to the algorithm's start. This style of hierarchical feature space exploration fits perfectly Algorithm 6 and its specific configurations.

From the perspective of Big Data, an introduction of some hierarchies of granularity into the spaces of investigated attributes can make the feature selection and extraction processes more efficient. Tackling the complexity of large data sets is an issue noticed by many researchers [107, 121]. The typical challenges associated with Big Data, as symbolized by the presence of "Five Vs", make things even more complicated. Besides the complexity and scale of calculations that affect the required amounts of resources, the superfluous features may negatively influence the understanding of the data by the analysts, therefore, affecting their ability to monitor and tune the knowledge discovery processes [179, 318].

Models and frameworks for parallel computing focus on various aspects of data processing [139]. Some of them respond to high velocity of the data, which makes them closer to incremental stream processing [144]. Others concentrate on batch processing models and adapt well-known mechanisms, such as the apriori-based breadth first exploration of a feature space [403]. Herein, the MapReduce paradigm

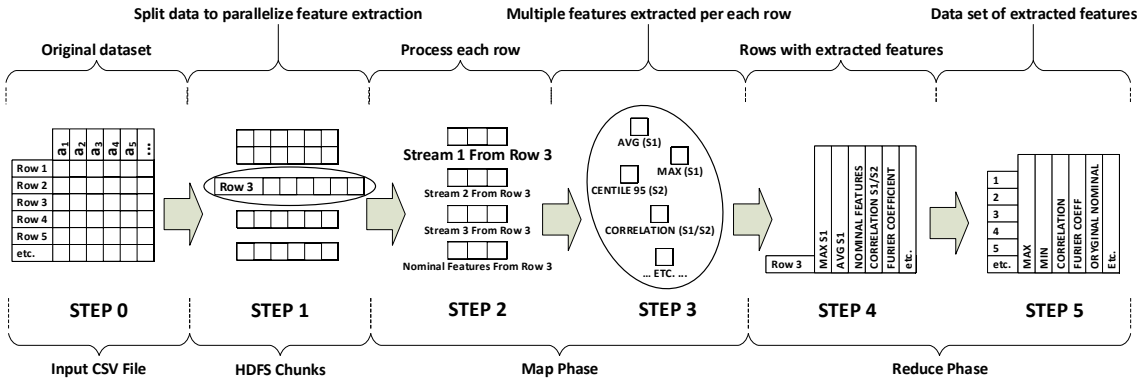


Figure 4.5: The overview of feature extraction process split into individual steps. The labels above the curly braces (at the top of the diagram) indicate objectives in each processing step. The labels below the curly braces (at the bottom of the diagram) indicate how the individual processing steps were implemented.

seems to be a good choice to consider [231, 236]. We may distinguish two popular approaches in this field. One of them implements the solution as a single job, whereas the other – iterative MapReduce – encompasses ℓ consecutive job runs that may be controlled automatically or manually [68, 105]. One can think about parallelization of the discussed granular feature selection methods using both of these approaches.

Let us outline one of possible implementations of a massively parallel granular feature selection process as an iterative MapReduce program. Consider ℓ consecutive iterations, where each of them is based on Algorithm 8. We propose to work on the transmuted data, i.e., the mappers are executed on attributes a assigned to a granule G_a and having a vector V_a of values for objects/records in the analyzed data set. The outcome of a single iteration is a sorted set of candidate attribute subsets, whereas only n best intermediate outputs $\mathbb{R} = \{R_1, .., R_n\}$ are passed to the subsequent phase. The map functions are provided with the collection $\mathbb{R}$ and the vector V_d containing values of the decision attribute d. To each subset R_i there has been assigned granulation preferences $\sigma^i_{\mathbb{G}}$, whereby the diversification of granule-level permutations may play a similar role as for the previously discussed attribute-level permutations. During the evaluation of a, we verify its relevance to every considered R_i, with respect to a quality function Q, preferences $\sigma^i_{\mathbb{G}}$, or any other factor of interest. If the performed assessment reveals that a is relevant for R_i (where relevance may be expressed as a mixture of preference, contribution, etc.), then the set $R_i \cup \{a\}$ is emitted. The role of reducers is then to aggregate subsets R_i and sort them according to their score. The whole process ends when the expected number of feature subsets satisfies $\mathbb{C}$.

The main objective of the above illustrative example of a MapReduce program is to evaluate a possibly large number of attribute subsets, in order to reach a higher quality, compactness and/or diversification of the produced outcomes. Obviously, parallel programming models allow to implement the granular feature selection framework in many other ways [236,307]. Let us also mention DiReliefF, a distributed version of the well-known ReliefF [283], or fast-mRMR algorithm for high-dimensional data [313].

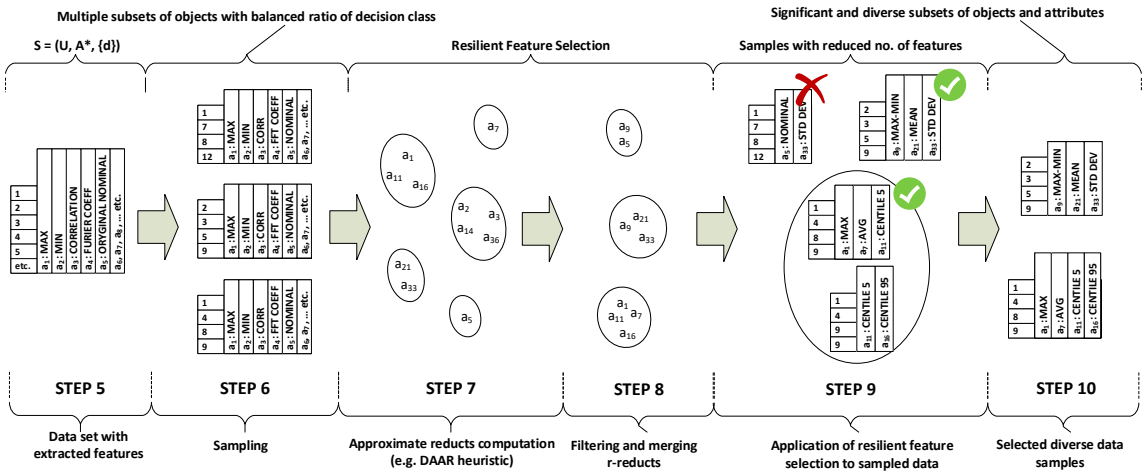


Figure 4.6: The overview of the feature selection process.

The idea of operating on attribute granules – regardless of their origin – is worth combining with the principles of parallelization of feature selection methods with respect to complex spaces of derivable features and their subsets, yet fitting the iterative nature of most ML algorithms [321]. In [346], granular computing was utilized to discretize M-factors time series data to obtain granular intervals. Information granules naturally emerge when dealing with data, including those coming in the form of data streams [293]. However, regardless the particular application the ultimate objective is to describe the underlying phenomenon in an easily understood way and at a certain level of abstraction to enable human-system interaction.

4.3 Framework for Multi-Stream Data Analysis

In this section, we focus not only on the extracted features and constructed prediction models but also on data processing stages that are designed to let it work within a big data processing environment [139, 312], and particularly with high dimensional, multi-stream data [181, 184]. In order to provide high quality assessments, the presented solution requires constructing an ensemble of diverse models [93, 179]. The diversity may be obtained by employing a variety of models computed on different subsets of attributes and data samples. For this task, the granular similarity measures (Section 4.2) or resilient attribute subsets (Section 3.2) may be applied. As a result of blending diverse models, the final ensemble minimizes the impact of a concept drift [44], and achieves a better prediction quality [179]. The proposed architecture can be used both in the incremental, stream processing model [36, 136], and in highly scalable, batch processing model, i.e., MapReduce [22, 87].

In Figure 4.5, a high-level overview of the feature extraction process divided into individual steps is presented. The 'original data set' in STEP 0 corresponds to a collection of historical data provided as a training set for a machine learning task, where features: $a_1, a_2, a_3, ..$ correspond to attributes in the data. STEP 1 is designed

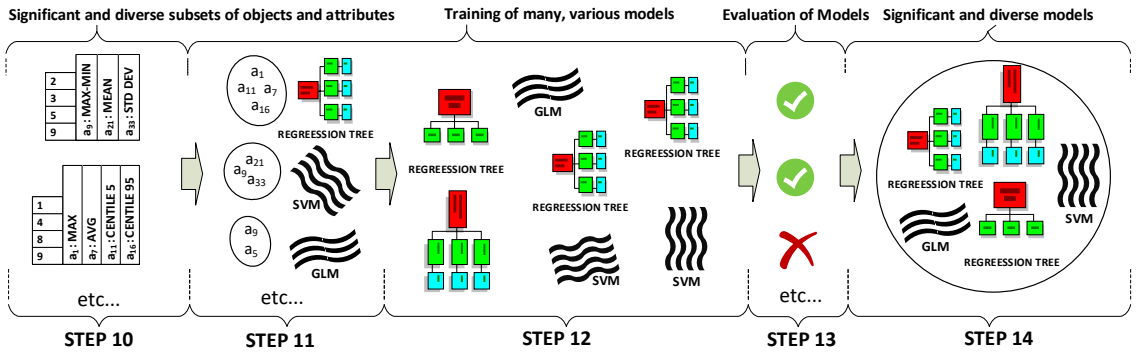


Figure 4.7: Ensemble blending.

to partition the original set into individual rows (objects) in order to parallelize calculations - this step may be implemented within, e.g., MapReduce framework [138]. The purpose of STEP 2 is to split each row into static data, e.g., features reflecting assessments obtained interactively from domain experts (as discussed in Sections 2.4, 4.2), and time series data (Section 4.1.2). In STEP 3, the feature extraction framework is applied to each time series in the data, e.g., to a numerical time series containing consecutive values expressing the average energy of the most active geophones at a longwall in a coal mine [179], and all features derived from time series are constructed (as described in Section 4.1.2). In STEPs 4 and 5, all attributes are combined together.

The process of feature engineering is performed basing on the sliding time-window approach that is designed to process data sets containing multiple time series (Section 4.1.2). During the process of moving a time window through the series, aggregating functions are applied. Table 4.1 presents the overview of features that may be extracted from individual time series. As emphasized in Section 4.1.2, standard statistics extracted form a sliding window may be supplemented by more sophisticated ones, e.g., correlations between pairs of time series. Furthermore, since more than one window is generated per time series, we may extract inter window statistics - as depicted in Figure 4.1. That is, a set of values which express changes between the same statistics obtained in consecutive sliding windows. The inter window stats are presented in Table 4.1.

During the feature extraction process, a large number of potentially relevant, however very often redundant, data characteristics are generated [143, 419]. Therefore, after the construction of features, an attribute subset selection algorithm is applied to reduce the attribute space [59]. In the case of designing decision support systems, the scope of feature selection is twofold, related to both interaction with domain experts and analysts while running the system on-line, as well as off-line exploration of gathered data in order to find out the best feature selection algorithms and prepare the best possible feature sets for further processing [55, 358]. It is herein worth noting that the step of feature selection – conducted independently or in an iterative fashion – is often taken into account in combination with various machine learning methods such as neural networks or support vector machines while building decision systems, e.g., aimed at equipment and environment monitoring in

coal mines [234].

In Figure 4.6, the overview of the feature selection process, split into individual steps, is presented. In STEP 6, the random samples of objects with a more balanced distribution of classes are drawn. The generated samples are randomly divided into two disjoint groups. First one serves for the purpose of feature selection, whereas the second group of samples is used for training predictive models. It should be noted that, in order to use the feature selection algorithms derived from RST (selected RST based FS methods are surveyed in Section 2.5, the novel ones are introduced in Chapter 3), numerical attributes in data should be subjected to discretization (discretization methods are surveyed in Section 2.1).

In STEP 7, the reduced attribute subsets are calculated. With regard to the proposed architecture (Figure 4.6), we decided to focus mainly on filter-based methods, which (comparing to wrapper and embedded techniques - Section 2.3) assure relatively high computational efficiency, as well as independence of the resulting feature sets from a particular model. This last property allows the obtained feature sets to be used in combination with various types of forecasting approaches. Among filter-based feature selection methods, we pay special attention to multivariate algorithms. One of the most prominent examples of this approach are methods based on the mRMR framework. Another popular approach that is implemented in the presented framework refers to computation of approximate decision reducts developed within the theory of rough sets, e.g., dynamically adjusted approximate reducts (DAAR), where a statistical test based on random probes is used to avoid selection of features that are likely to distinguish data records supporting different target classes only by chance. In STEPs 8 and 9, a number of feature subsets computed in the previous step are merged into several larger subsets. For this purpose, we refer to the presented version of the approximate resilient feature selection Algorithm 5. In STEP 10, only significantly different attribute subsets are maintained for the purpose of model training.

In order to provide a good clarity of the presentation, in the subsequent steps of the framework (in Figure 4.7), let us focus on the well known task of regression analysis[1]. The final solution is an ensemble of diverse regression models. The diversity is achieved by training models on different subsets of attributes and objects (STEP 11). The Algorithm 9 for blending models refers to STEPs: 11 to 14 in Figure 4.7, and is designed in a way which guarantees that a model can be included only if it is accurate enough on validation data, and sufficiently different from already selected predictors (e.g., correlation of its prediction with predictions of other models is small enough). This could be seen as a method for increasing robustness of predictions in the case of noisy and heterogeneous data. An additional advantage of using different features for different models is that it may reduce the influence on the final ensemble of a concept drift between the training and test cases. This approach is also expected to protect the model against over-fitting and, hence, a decrease in the prediction quality on the final test set.

[1]The description for the classification tasks would differ mainly in the training algorithms, and model evaluation criteria used.

Algorithm 9: Construction of the ensemble of diverse regression models.

Data:
- *attSubsets* - pre-calculated subsets of attributes (e.g., approximate reducts)
- *objectSamples* - pre-calculated samples of objects
- *testSet* - a test set
- *regressionAlgorithms*, e.g., { rPart, SVM, glm }
- *allowedAttempts* and *minQuality* - parameters governing quality of models

Result: *ensemble of regression models*

```
/* Initialization of variables                                          */
ensemble ← ∅; weakAttempts ← 0
alg ← regressionAlgorithms.removeFirst
while TRUE do
    a_1, a_2 ← attSubsets.drawAndRemoveTwo
    b_1, b_2 ← objectSamples.drawAndRemoveTwo
    /* Models are trained and validated on different samples            */
    model ← alg.trainAndEvaluate(a_1, b_1, a_2, b_2)
    score ← model.score(testSet)
    /* The ensemble is expanded only if the newly trained model meets
       the specified quality criteria and there is no other similar
       model in the ensemble.                                           */
    if score > minQuality ∧ ¬ensemble.containsSimilar(model, score) then
        ensemble ← ensemble ∪ {model ⊕ score}
    else
        weakAttempts ← weakAttempts + 1
        if weakAttempts < allowedAttempts then
            continue;
        end
        if regressionAlgorithms ≠ ∅ then
            alg ← regressionAlgorithms.removeFirst
            weakAttempts ← 0
        else /* end of ensemble blending */
            break;
        end
    end
end
return ∑_{s∈ensemble.scores} s;
```

Chapter 5

Evaluation, Practical Applications

The goal of this Chapter is to present examples of a data mining investigation, which illustrate the performance of the framework presented in Chapter 4 and the resilient feature selection methods described in Chapter 3 when handling real-life problems related to coal mining and fire & rescue domains. The following study complements the so-far presented research with the experimental evaluation of the proposed feature extraction and selection methods.

5.1 Methane Outbreaks

In this section, we provide a broad experimental evaluation of learning forecasting models over large multi-sensor data sets, including the steps of sliding window feature extraction and rough-set-inspired feature subset ensemble selection. The considered task is to construct a model capable of predicting dangerous concentrations of methane at longwalls of a coal mine basing on multivariate time series of sensor readings. We show how the described framework performed on data collected from a sensor network in an active coal mine and, how the complete mechanism can be built into DISESOR - a particular decision support system.

The contributions in this section refer to both the analysis of how the nature of sensor readings influenced the architecture of the developed solution and the empirical proof that the designed methods turned out to be efficient in practice. Furthermore, we elaborate on the resilience of the solution in the case of partial data loss, e.g., when particular data sources (e.g., sensors) are damaged or inactive.

5.1.1 Natural Hazards Monitoring in Coal Mines

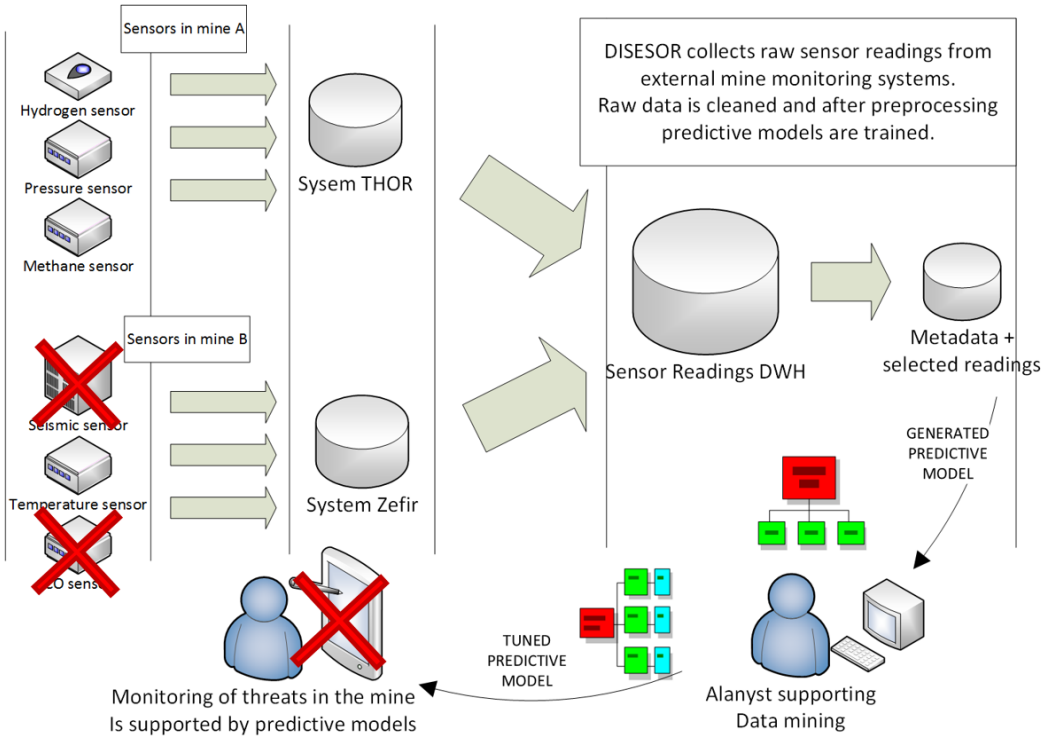
Coal mining requires working in hazardous conditions. Miners in an underground coal mine can face several threats, such as methane explosions, rock-bursts or seismic tremors, etc. [48, 234, 257]. To provide protection for people working underground, systems for active monitoring of production processes are typically used [216]. One of their fundamental applications is screening dangerous gas concentrations (methane in particular) in order to prevent spontaneous explosions [181]. For that purpose, the ability to predict dangerous concentrations of gases in the nearest future can be even more important than monitoring the current sensor readings [356].

Coal mines are well equipped with monitoring, supervising, and dispatching systems connected with machines, devices, and transport facilities. There are also systems for monitoring natural hazards. Such systems are provided by many different companies, hence some problems with data quality, integration, and interpretation may be observed (Section 4.1.1). Once someone is able to overcome these issues, the collected data can be used for ongoing visualization of conditions in particular places of a mine [271]. Moreover, by utilizing the domain knowledge and patterns derived from integrated historical data [113], one can construct forecasting models to enrich the upcoming sensory data with additional predictions. This way, it is possible to considerably improve both the safety of miners and work efficiency [265]. For example, thanks to short-term prognoses related to methane concentrations combined with information regarding the location and work intensity of a cutter loader, it is possible to prevent emergency energy shutdowns and maintain continuity of mining [181]. This, in turn, allows for increasing the production volume and reducing the wear of electrical elements whose exploitation time largely depends on a number of switch-ons and switch-offs [345]. Furthermore, a decision support system should be easily comprehended by the experts and end-users who, not only, need access to its outcomes, but also to arguments or causes that were taken into account (Section 4.2).

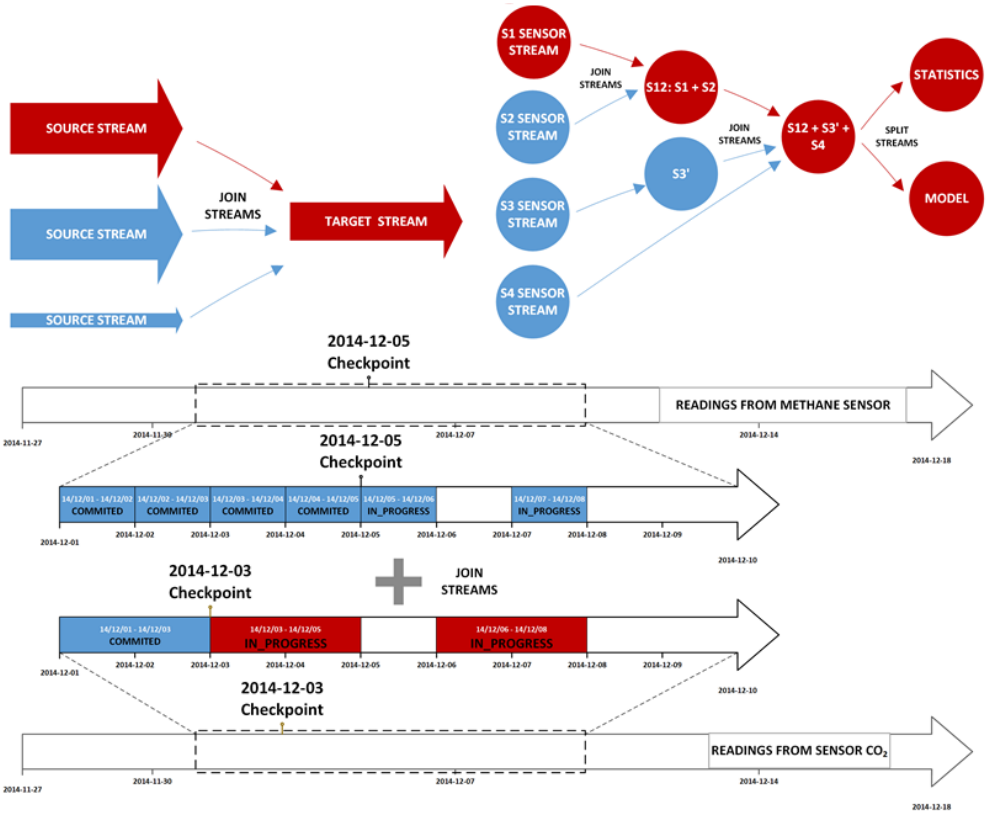
DISESOR is a decision support system designed for monitoring potential threats in coal mines [356], which processes data from sensors of various types, like: CO2, methanometers, machine monitoring devices and many others (Table A.1, and Appendices: A.1, A.2). In Figure 5.1 (a), a draft architecture of DISESOR system is presented [303]. Basing on a vast amount of sensor readings collected via integration of data from various monitoring systems [140], e.g., THOR or Zefir, DISESOR provides predictive analytics of mine conditions and threats. As the most important use cases of the system we can indicate: Assessment of seismic hazard probabilities in the vicinity of the mine; Forecasting dangerous increase in the methane concentration in the mine shafts; Forecasting of the possible ranges of the sensor readings in advance; Detection of endogenous fires and conveyor belts fires; Detecting anomalies in the consumption of media such as electricity [178].

From data processing point of view, a decision support system that aid in controlling the coal mining process requires efficient methods that can handle large volumes of data from many sensors enriched with features provided by domain experts (Sections 4.2.3, 4.3). The continuous collection and analysis of multiple streams of readings from a large network of sensors located underground raises certain problems with providing expected resilience (Chapter 3) in business continuity plans. In Figure 5.1, a potential impact of missing data sources is shown. The bottom drawing in Figure 5.1 (b) outlines – in a schematic manner – the window based processing of two data streams from methane and CO_2 sensors. We may notice on the drafts above that the impacted stream (in red) propagates the problem forward and blocks the subsequent processing tasks. That is especially harmful when we combine a number of sources – i.e., joining data streams in Figure 5.1 (b). The majority of predictive models are very sensitive to the (in)completeness of input data.

Predictive models providing a proper assessment of potentially dangerous methane concentrations (Section 5.1) and seismic events (Section 5.2), which are resilient



(a)



(b)

Figure 5.1: DISESOR system architecture (a) and sliding window data flow (b).

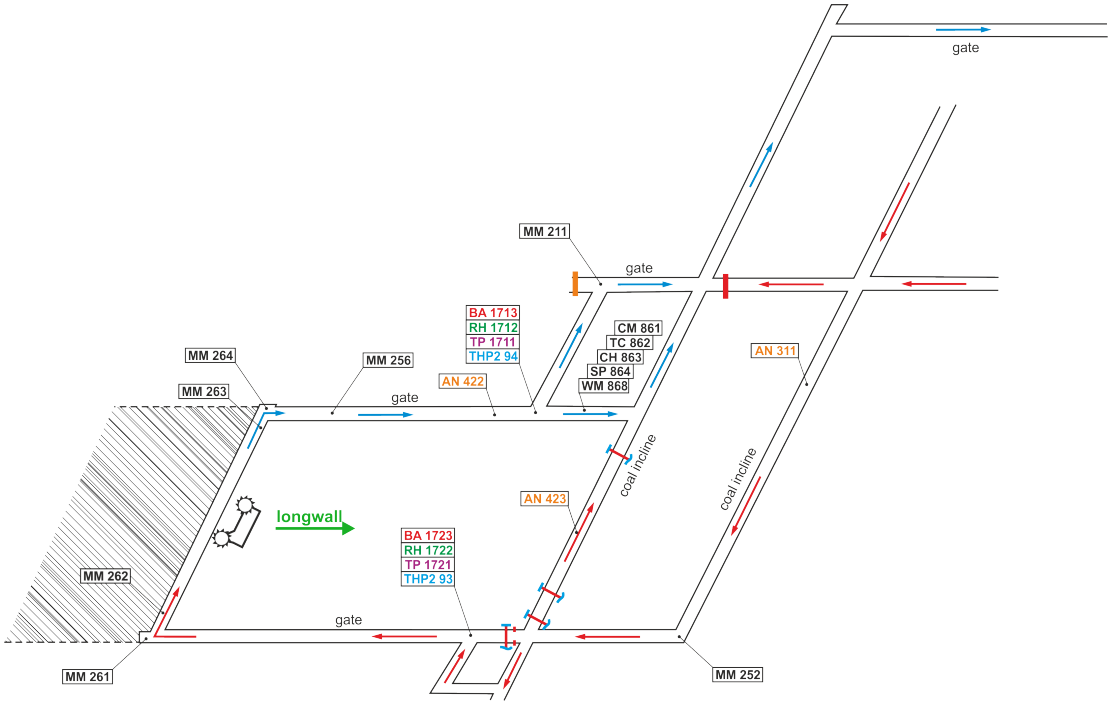


Figure 5.2: A scheme of the mining process corresponding to the data set considered in [181]. A shearer moves along the wall of coal extraction between the sensors MM261 and MM264. The progress of the coal extraction is unveiled by an arrow described with "longwall". Thin arrows depict flow direction of the air in the mine sidewalks which is enforced by a ventilation system.

to missing data sources and are able to interact with domain experts to use their assessments, could significantly improve the safety and reduce the costs of underground coal mining.

5.1.2 IJCRS'15 Data Challenge

Based on the sensor readings collected in an active Polish coal mine, a data mining competition was organized at the international conference IJCRS'15 [181]. By publishing this data set and defining the corresponding problem in the form of a competition task, we obtained and analyzed 1,676 solutions submitted by 90 registered research teams from 18 different countries. Additionally, 40 teams provided reports describing their approach. Altogether, these solutions can be regarded as state-of-the-art in the predictive analysis of multivariate time series data and as the reference in our research.

Data prepared for the competition correspond to a mining period between March 2, 2014, and June 16, 2014. Among the thousands of sensors located over tens of kilometers of underground corridors, 28 sensors monitoring the work in the immediate vicinity of the shearer workplace were selected. Prepared data records were composed of raw sensor readings arranged in 10-minutes time series, with measurements taken every second. Hence, each record consisted of 16,800 numerical features, i.e., 600 values per sensor. The detailed information about all the sensors can be found in

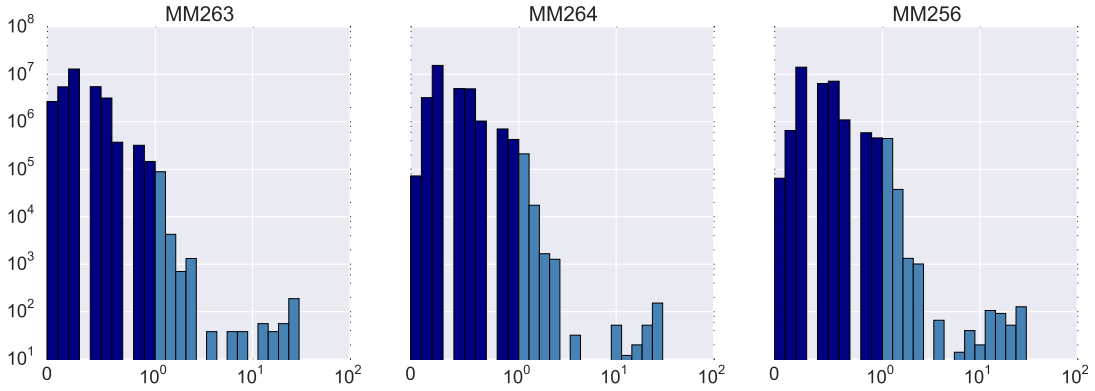


Figure 5.3: Frequency distributions for sensors MM263, MM264 and MM256 in the training data set. The majority of readings are in the $[0, 1.2]$ range and a relatively small number is spread in the $(1.2, 30]$ range. The dark blue bars drawn on a linear scale with a step of 0.1 correspond to the readings below the warning threshold the threshold $\rho = 1\%$. The light blue bars drawn on a logarithmic scale with a step of 1 represent hazardous situations as well as outliers.

Table A.1. In Figure 5.2, a detailed location of all sensors, as well as a workplace of a longwall shearer, on a fragment of the coal mine plan is shown.

Each row of the training data was tagged with three labels, each from the set $\{0, 1\}$, where 0 and 1 corresponded to "normal" and "warning" labels, respectively. Labels indicated whether a warning threshold had been reached in a period between three and six minutes after a given measurement, for three methane meters denoted as $MM263$, $MM264$ and $MM256$ (Figure 5.2). If a given data row corresponded to a time period between t_{-599} and t_0, then its dependent variable value for a meter MM was "warning", if and only if $\max\{MM(t_{181}), \ldots, MM(t_{360})\} \geq \rho$, where ρ is a safety threshold. The value of this threshold may vary for different longwalls, however, it is usually set between 1% and 1.5% on the basis of interviews with experts (see, e.g., a sensor card in [356]) and the national regulations on hazard estimation [135].

The training set contained sensor readings registered within $51,700$ time periods. Periods in the training set were overlapping and given in a chronological order. However, periods included in the testing set did not overlap and they were given in a random order. Figure 5.3 presents frequency distributions of values for the three sensors $MM263$, $MM264$ and $MM256$. The vast majority of the observations stored in the data set are below the warning threshold. Table 5.1 presents the amount of "warnings" observed for each investigated sensor in the training data and may be used as a premise to realize the decision class imbalance in the context of methane concentration monitoring. Selected, more in-depth insights into the methane-related data are provided in Appendix A.1. Data sets are available online on the KnowledgePit platform.

The task of the data challenge was to predict the likelihood of the label 1 ("warning") for the threshold $\rho = 1\%$. The solutions were evaluated with the Area Under the ROC Curve (AUC) measure, which was computed separately for each of the target sensors. The final score corresponded to the average AUC for a submitted solution "s":

Table 5.1: Occurrences of the labels in the training data set.

	MM263	MM264	MM256	count
label values	normal	normal	normal	48695
			warning	1208
		warning	normal	1258
			warning	74
	warning	normal	normal	435
			warning	24
		warning	normal	2
			warning	4

$$score(s) = \frac{AUC_{MM263}(s) \; + \; AUC_{MM264}(s) \; + \; AUC_{MM256}(s)}{3} \tag{5.1}$$

Table 5.2 shows the scores of top-ranked teams together with the score of our methods. Baseline model was created by averaging 10 simple rule-based models computed out-of-the-box using the *RoughSets* package [319]. The Zagorecki approach [418] assumed generation of a large number of variables characterizing sensor measurements and operating with the time series derived from those measurements. Separate random-forest-based models were then created to predict the "warning" states for each of three considered methane meters. On the other hand Boullè [43] focused on a problem of distribution drift between the training and testing data sets. Informativeness of each considered feature with respect to both classification and drift detection was evaluated. As a result, the training data set was reduced to a single sensor per target class. The prediction model was then generated by the Naïve Bayes classifier. Grzegorowski and Stawicki [143] provided a logistic regression model based on the linear combination of selected three features – extracted with the sliding window framework (presented in Section 4.1.2). The Ruta and Cen [327] method was also based on a logistic regression model computed over a small subset of sensor observations. The authors utilized their self-organizing framework to choose this particular model out of a number of other solutions including decision trees, support vector machines, etc. Among other successful approaches used by participants of the competition, there were also deep learning models using the LSTM networks [289].

Based on the analysis of the most successful solutions submitted to the competition (Table 5.2), we reached a conclusion that a robust prediction of methane concentration levels can be achieved even when a small subset of features is used for constructing the model. Although the Zagorecki [418] solution used nearly $5,000$ features in the learning process, several of the other top-ranked teams achieved similar results with models considering far fewer features. Another interesting outcome was that a vast majority of solutions followed the ideas of producing sliding window aggregations and that such aggregations, treated as low-level features, were useful while learning various prediction models [223].

Table 5.2: A comparison of the logistic regression performance for the competition data set between the implemented feature extraction methods and the top ranked solutions of IJCRS'15 Data Challenge.

Method	Macro averaged AUC (5.1)
Zagorecki [418]	0.9592
FE+DAAR	0.9545
Grzegorowski and Stawicki [143]	0.9473
Boullè [43]	0.9439
Ruta and Cen [327]	0.9436
FE+mRMR	0.9413
Pawłowski and Kurach [289]	0.94
Baseline	0.9004

5.1.3 Evaluation of Multi-Stream Framework

In the second part of our experiments, we considered exactly the same sensor readings as those used in Section 5.1.2, now, taking into account our multi-stream feature extraction framework with ensemble blending (Section 4.3). We applied two different feature selection methods into our framework and we examined the AUC scores of prediction outcomes obtained using ensembles of simple models. Both cases were following the general framework presented in Section 4.3, in particular, in Algorithm 9. In both cases the training algorithm was used independently for each of three dependent variables – implementing a particular transformation for the original multi-target problem [41]. Subsequently, we extended the analysis with two new multi-stream data sets, both related to hard coal mining, to verify how effectively the discussed approach could transfer to new data in the same domain with implementation changes limited to adaptation of the sliding window feature extraction layer only [330].

In the first setup of the framework, we applied our version of the minimum redundancy maximum relevance (mRMR) method [313]. Comparing to the standard mRMR, provided modifications are related to criteria for selecting the best feature in each iteration and to a stopping condition – outlined in Algorithm 10. First, we select a feature that maximizes the difference between its relevance (the dependency score $\phi(a, d)$) and its maximal dependency on features selected before. Second, we stop the algorithm if the feature selected in a given iteration does not pass the random probe test, i.e., the estimation of the probability that a randomly generated feature obtains a higher score than the selected feature exceeds an allowed threshold [368]. Thus, we guarantee compactness and the relatively high independence of the resulting feature set.

We obtained three small subsets of features containing three, six and seven elements, respectively. With these feature sets, we trained three independent logistic regression models and utilized them to make predictions for the testing cases. Although we used a very simple prediction method and a completely automated feature selection, the average AUC of this solution for the testing set was 0.9413 – see FE+mRMR in Table 5.2.

In order to verify the stopping criteria we repeated the experiment with the probe

Algorithm 10: Implemented version of mRMR feature selection method.

Input: set of features A and dependent variable d;
$\phi : A \times A \cup \{d\} \to \mathbb{R}^+$ function for measuring dependency;
$N \in \mathbb{N}$; $\varepsilon \in [0, 1)$;
Output: subset of features $R \subseteq A$

begin
  $stopFlag \leftarrow FALSE$;
  $R \leftarrow \arg\max_{a \in A} \ \phi(a, d)$;
  $A \leftarrow A \setminus R$;
  while $stopFlag == FALSE$ **do**
    $\bar{a} \leftarrow \arg\max_{a \in A} \ \Big(\phi(a, d) - \max_{b \in A'} \ \phi(a, b)\Big)$;
    foreach $i \in 1, ..., N$ **do**
      $\bar{p}_i \leftarrow$ random permutation of $\bar{a}$;
    end
    if $\frac{|\{i : |\phi(\bar{p}_i, d)| > |\phi(\bar{a}, d)|\}| + 1}{N+2} > \varepsilon$ **then**
      $stopFlag \leftarrow TRUE$;
    else
      $R \leftarrow R \cup \bar{a}$
    end
  end
end

condition switched off. Figure 5.4 shows the results of the forecasting model trained using features selected in 25 consecutive iterations of the mRMR procedure. It can be seen that, for each of dependent variable values, the results obtained for our stopping method were close to optimal. Moreover, by virtue of the random probe test, the resulting feature subsets were compact. Overall, they contained the lowest number of features among the solutions submitted by the top-ranked teams in the competition.

In the second setup, we implemented the granular algorithmic schema outlined in Algorithm 7 using – as a granulation technique – a DAAR heuristic [183]. For each of three dependent variables, we computed 10 different feature sets. Then, for each set, we computed a logistic regression model using the same training algorithm as for features selected with mRMR. Finally, we created a simple ensemble by averaging their prediction. The final score achieved by the ensemble of DAAR-based logistic regression models was 0.9545 – see FE+DAAR in Table 5.2.

In order to thoroughly investigate the performance of the proposed framework with particular interes on feature extraction and selection part, we evaluated it on two additional data sets that were obtained from different coal mines. The first of those sets contained sensor readings from over a month (November 2007 – December 2007). Similarly as the first set there were three target methane meters (MM532, MM533 and MM534) and the sensor reading frequency was one per second. The data set contained readings from eight sensors plus additional information regarding the coal shearer status. After the initial preprocessing, the data consisted of 51,329 records corresponding to 10-min periods of sensor readings. These records were divided into two disjoint sets - one for training the compared models and the second one for

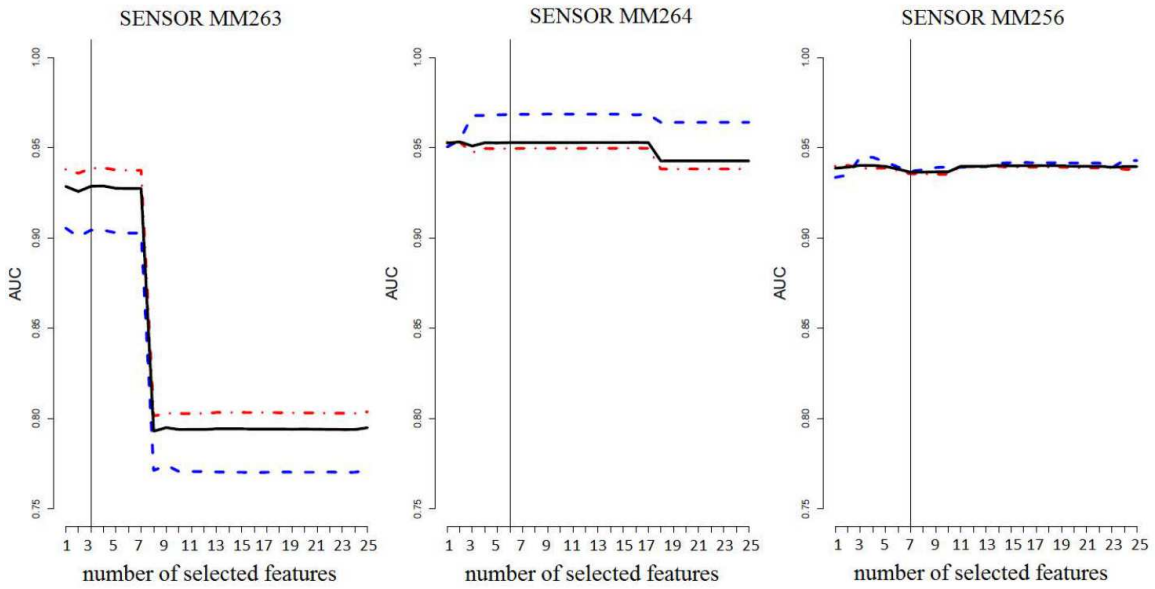


Figure 5.4: AUC values obtained for simple linear regression models trained on features selected in subsequent iterations of mRMR procedure for each of dependent variable values in the data: dashed blue lines show the scores on the preliminary testing set; dashed-dotted red lines show the scores on the final testing set; thick black lines show the scores on the joined testing set; thin vertical lines mark the iteration on which the stopping condition of our implementation of mRMR algorithm was triggered.

validation. The testing set corresponded to the last two weeks of sensor readings.

In the second of the considered new data sets, there was only one target methane meter (denoted by MWR116) and the data spanned across five months (September 2013 – January 2014). In this set, there were readings from 14 sensors, sampled once per minute. After the initial preprocessing, the set consisted of 204,465 records. They were divided into separate training and testing sets as well. The test period corresponded to the last two months of sensor readings. The results obtained for each of the target sensors from all three data sets are presented in Table 5.3. In addition to our own models, we include there the results obtained using implementation of the model reported by the IJCRS'15 competition winner [418].

It is also worth to notice that the multi-stream framework setup with DAAR feature selection algorithm achieved the highest macro average AUC on all seven target sensors (Table 5.3). The paired Wilcoxon test did not reveal statistically significant differences in the results between the AutoML multi-stream framework results (Section 4.3) and the best of the fine-tuned solutions constructed on nearly 5000 features. It is, therefore, sufficient to conclude that the proposed framework can successfully replace more complicated and hardly interpretable machine learning approaches. Moreover, computation time required to train our model was an order of magnitude lower. For instance, for the third data set, our model was constructed in 19 min, whereas the construction of the random forest model took nearly five hours.

The aforementioned analysis clearly shows the accuracy obtained using our approach, taking into account its subsequent layers of feature creation and selection (Figures 4.5 and 4.6) and the forecasting models training and ensemble blending

Table 5.3: A comparison of logistic regression performance (AUC measure) for individual target methane meters from the three data sets considered in our study (macro averaged AUC – in the last row).

Target variable	FE+mRMR	FE+DAAR	Zagorecki [418]
MM256	0.9432	**0.9579**	0.9439
MM263	0.9374	0.9564	**0.9760**
MM264	0.9433	0.9492	**0.9579**
MM532	**0.9176**	0.9170	0.8968
MM533	0.8501	**0.8681**	0.8283
MM534	0.9276	**0.9321**	0.9299
MWR116	0.9389	0.9431	**0.9575**
Macro Averaged AUC	0.9226	**0.9320**	0.9272

(Figure 4.7). Both evaluated feature selection approaches – mRMR and DAAR methods – yield very good results even when combined with the simplest possible prediction techniques – the logistic regression. From the prediction accuracy perspective, they perform comparably to the model developed by the competition winner which was manually tuned for over two months. Moreover, they are easy to maintain, efficient to compute and, what is maybe the most important aspect from the point of view of interactiveness, they are understandable for the system users and domain experts by means of operating with small subsets of intuitively defined features. In particular, this is why the DAAR-based method was deployed in a production system responsible for processing sensor readings collected from multiple monitoring and dispatching systems deployed in different coal mines. [356].

5.1.4 Impact of Feature Extraction on Resilience

Let us evaluate the impact of the sliding window-based feature extraction on the quality and resilience of methane prediction. For that purpose, we refer to the same data set of sensor readings collected from an active coal mine in Poland between March 2, 2014, and June 16, 2014 - as described in Section 5.1.2. In the following study, we performed a series of experiments on both raw, unprocessed data and on the data after performing feature extraction (as described in Chapter 4). The problem was to predict maximal methane concentrations in a six minutes time horizon for three selected methane meters, denoted as: $MM263$, $MM264$ and $MM256$ (Figure 5.2) – similarity as in JCRS'15 Data Challenge – see Section 5.1.2. This time, however, evaluations were performed with three error measures designed to assess regression problems: mean absolute error (MAE), root mean squared error (RMSE), and root relative squared error (RRSE).

In the frame of the experiment, we focused on those methods that can predict the methane concentration even in the absence of selected conditional attributes, including linear regression (lm), two implementations of regression trees (rpart, ctree), regression rules (cubist), and gradient boosting (gbm). As in most short-term regression problems, the last know value (last val.) is usually a reasonably good naive approach, commonly used as a baseline forecast. Apart from that, we also used as

Table 5.4: Impact of missing features on prediction quality. No feature extraction applied.

Target Attr.	Missing:	-			MM256			MM263, MM264			MM256, MM263, MM264		
	Method	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE
MM256	train μ	0.1756	0.3532	1	0.1756	0.3532	1	0.1756	0.3532	1	0.1756	0.3532	1
	last val	**0.0733**	0.2971	0.841	-	-	-	**0.0733**	**0.2971**	**0.841**	-	-	-
	lm	0.1335	0.3198	0.9052	0.2166	0.3834	1.0855	0.0969	0.3033	0.8586	0.19	0.3652	1.0339
	rpart	0.0906	**0.2941**	**0.8327**	0.1688	0.3796	1.0747	0.1179	0.7232	2.0474	0.1846	0.5273	1.4928
	ctree	0.0918	0.3297	0.9334	**0.144**	**0.3395**	**0.9611**	0.0883	0.305	0.8635	**0.1529**	**0.3491**	**0.9883**
	gbm	0.1623	0.3459	0.9793	0.1719	0.3508	0.993	0.1623	0.3459	0.9792	0.1738	0.3516	0.9955
	cubist	0.1328	0.3892	1.1017	0.2198	0.4047	1.1456	0.1238	0.3384	0.958	0.3555	0.632	1.7891
MM263	Missing:	-			MM263			MM256, MM264			MM256, MM263, MM264		
		MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE
	train μ	0.1331	0.3247	1	**0.1331**	**0.3247**	**1**	0.1331	0.3247	1	**0.1331**	**0.3247**	**1**
	last val	**0.0498**	0.2909	0.8957	-	-	-	**0.0498**	**0.2909**	**0.8957**	-	-	-
	lm	0.0612	**0.2905**	**0.8945**	0.176	0.3484	1.073	0.0601	0.291	0.8961	0.1772	0.3483	1.0725
	rpart	1.1592	3.0303	9.3317	0.1348	0.3285	1.0115	1.1592	3.0303	9.3317	0.1337	0.3263	1.005
	ctree	0.2502	1.2467	3.8391	0.1792	0.3577	1.1015	0.2508	1.246	3.8369	0.1826	0.3591	1.1058
	gbm	0.1245	0.3198	0.9849	0.1341	0.3255	1.0024	0.1245	0.3198	0.9849	0.1341	0.3255	1.0023
	cubist	0.1796	0.6836	2.105	0.2517	0.4375	1.3474	0.1005	0.331	1.0192	0.1869	0.3683	1.134
MM263	Missing:	-			MM264			MM256. MM263			MM256. MM263. MM264		
		MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE
	train μ	0.165	0.3267	1	0.165	0.3267	1	0.165	0.3267	1	0.165	0.3267	1
	last val	**0.0632**	**0.2704**	**0.8277**	-	-	-	**0.0632**	0.2704	0.8277	-	-	-
	lm	0.0823	0.2714	0.8308	0.165	0.3267	1	0.0715	**0.2667**	**0.8165**	0.1711	0.3316	1.0151
	rpart	0.0738	0.3411	1.0441	0.1799	0.341	1.044	0.0819	0.3981	1.2187	**0.1534**	**0.3217**	**0.985**
	ctree	0.0892	0.3231	0.9892	0.2328	0.413	1.2643	0.0722	0.269	0.8236	0.2256	0.3857	1.1808
	gbm	0.1551	0.3192	0.9773	**0.1623**	**0.3247**	**0.9939**	0.1552	0.3192	0.9773	0.1636	0.3249	0.9946
	cubist	0.0857	0.2894	0.886	0.4085	0.6752	2.0671	0.1056	0.303	0.9276	0.2559	0.4242	1.2986

Table 5.5: Impact of sliding window feature extraction on resilience of methane prediction.

Target	Missing:	-			MM256			MM263, MM264			MM256, MM263, MM264		
Attr.	Method	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE
MM256	rpart without FE	0.0906	0.2941	0.8327	0.1688	0.3796	1.0747	0.1179	0.7232	2.0474	0.1846	0.5273	1.4928
	rpart+FE(l:1/o:1)	0.0713	0.2661	0.691	0.1146	0.3126	0.7248	0.0567	0.274	0.6888	0.1139	0.3067	0.7711
	rpart+FE(l:3/o:1)	0.0737	0.2543	0.6603	0.1163	0.2905	0.7542	0.0591	0.2897	0.7284	0.1204	0.316	0.8206
	rpart+FE(l:6/o:1)	0.0733	0.2582	0.6491	0.1141	0.224	0.5817	0.0701	0.2945	0.7403	0.1153	0.2628	0.6823
	best rpart+FE	**0.0713**	**0.2543**	**0.6491**	**0.1141**	**0.224**	**0.5817**	**0.0567**	**0.274**	**0.6888**	**0.1139**	**0.2628**	**0.6823**
	best without FE	0.0733	0.2941	0.8327	0.144	0.3395	0.9611	0.0733	0.2971	0.841	0.1529	0.3491	0.9883
MM263	Missing:	-			MM263			MM256, MM264			MM256, MM263, MM264		
		MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE
	rpart without FE	1.1592	3.0303	9.3317	0.1348	0.3285	1.0115	1.1592	3.0303	9.3317	0.1337	0.3263	1.005
	rpart+FE(l:1/o:1)	0.0521	0.2708	0.7395	0.137	0.3335	0.911	0.0457	0.2598	0.686	0.1234	0.3161	0.8348
	rpart+FE(l:3/o:1)	0.0753	0.2368	0.6467	0.1356	0.2862	0.7816	0.0544	0.2569	0.6785	0.1204	0.316	0.8206
	rpart+FE(l:6/o:1)	0.0521	0.2596	0.6459	0.12	0.2438	0.6658	0.0511	0.2526	0.667	0.1338	0.3004	0.8204
	best rpart+FE	0.0521	**0.2368**	**0.6459**	**0.12**	**0.2438**	**0.6658**	**0.0457**	**0.2526**	**0.667**	**0.1204**	**0.3004**	**0.8204**
	best without FE	**0.0498**	0.2905	0.8945	0.1331	0.3247	1	0.0498	0.2909	0.8957	0.1331	0.3247	1
MM264	Missing:	-			MM264			MM256, MM263			MM256, MM263, MM264		
		MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE	MAE	RMSE	RRSE
	rpart without FE	0.0738	0.3411	1.0441	0.1799	0.341	1.044	0.0819	0.3981	1.2187	0.1534	0.3217	0.985
	rpart+FE(l:1/o:1)	0.0698	0.2555	0.665	0.1227	0.2886	0.7337	0.0635	0.3262	0.8492	0.1257	0.328	0.8537
	rpart+FE(l:3/o:1)	0.07	0.2351	0.5978	0.1295	0.2651	0.6901	0.0497	0.2894	0.7533	0.1245	0.2925	0.7614
	rpart+FE(l:6/o:1)	0.0719	0.2433	0.6333	0.1241	0.3113	0.8102	0.0619	0.2952	0.7684	0.1294	0.2613	0.68
	best rpart+FE	0.0698	**0.2351**	**0.5978**	**0.1227**	**0.2651**	**0.6901**	**0.0497**	0.2894	**0.7533**	**0.1245**	**0.2613**	**0.68**
	best without FE	**0.0632**	0.2704	0.8277	0.1623	0.3247	0.9939	0.0632	**0.2667**	0.8165	0.1534	0.3217	0.985

a predictor a very simple statistic, that is an average calculated on the training set (train μ). Since the purpose of the experiment was to assess the relative impact of the performed feature extraction vs. using raw data, we did not enforce any parameter tuning of the models. The results are summarized in Table 5.4.

The first column in Table 5.4 indicates the target variable. The second one provides information about the prediction method used. Three consecutive columns provide information about the prediction error under the assumption that all the sensors from the training set were available during the assessment - the best results per each target variables in bold. We may notice that the results of simple approaches like the last value were quite often the best. Regression threes (rpart) performed very well in the case of $MM256$, however by far the worst in the case of $MM263$ – we could observe similar behavior whenever the $MM263$ was available in the conditional attribute set - evidently, the cause of the observed over-fitting. This raised the question of whether the rpart model on different data representation (i.e., with sliding window-based feature engineering applied) could result in a more robust behavior of this method.

Due to hazardous events or harsh conditions prevailing in mines, sensors or wires transmitting data may be damaged. As presented in Figure 5.1, this may cause gaps in the collected readings, resulting in missing values of particular attributes. Thus, all the predictive models utilizing affected features from failed sensors may become impacted (Figure 5.1). To verify the resilience of the methods to missing attributes, we performed the following three experiments: we had been removing from the test data the most important attributes: $MM256$, $MM263$, and $MM264$. The *"Missing:"* keywords in Table 5.4 indicate rows with sensor symbols that were excluded from the conditional attributes in the test set. Columns 6–8 indicate the error measured when the historical values of the target variable were not present - such a situation obviously disables forecasts with the *last value* predictor. The last three columns (12–14) provide an assessment of the methane concentration forecasts when all $MM256$, $MM263$ and $MM264$ sensors would be, for some reason, disabled. We may notice that tree-based methods handled relatively well that crisis-scenario.

In Table 5.5, we provide results of the same experiments, this time, however, performed on a slightly different representation of data - after the sliding window feature extraction was applied. For that study, we picked one of the tree-based models - *rpart* - and applied three different sliding window setting with respect to window length and offset (Recall, e.g., Figure 2.1) designated with "+FE" in Table 5.5. For each $rpart{+}FE(l:\ldots/o:\ldots)$ method, information in brackets indicates a particular sliding window setting, where "l:" indicates *a length of a time window* (in minutes) and "o:" refers to *an offset of a time window* (in minutes). To emphasize the quality gain achieved with sliding window feature extraction, we additionally added "rpart without FE" to the comparison. The last two rows per each target variable contain the best results achieved with feature extraction - that is $rpart{+}FE = MIN(rpart{+}FE(l:1/o:1), rpart{+}FE(l:3/o:1), rpart{+}FE(l:6/o:1))$ vs. the best results without FE - that is with minimal error in Table 5.4. We may notice that rpart with feature extraction outperformed the best of the evaluated methods 33 out of 36 times - the best results in bold.

The results confirm that the developed feature extraction methods have a positive

impact on prediction of methane concentration. An important contribution of this research is the evaluation of the impact of the developed feature extraction methods not only on the quality of prediction but also on the resilience of various machine learning models in the case of partial data loss, i.e., missing attributes. The performed experimental study confirms that it is feasible to assure a proper resilience level of methane concentration prediction. Hence, would allow us to immunize decision support systems in case of data loss.

5.2 Seismic Events

In this section, we investigate how the interactive feature extraction and ensemble blending methods, proposed in Chapter 4, generalize to other problems of multi-stream data analysis. Once again, we address the problem of safety monitoring in underground coal mines. This time, we investigate and compare practical methods for the assessment of seismic hazards using analytical models constructed on both multi-stream sensory data and features derived from domain experts. The possibility to represent a problem related to data exploration and analysis with machine-generated features enriched with expert assessments, we consider as one of the essential aspects from the point of view of interactiveness.

For our case study, we use a rich data set collected during a period of over five years from several active Polish coal mines. We focus on comparing the prediction quality between expert methods, which serve as a standard in the coal mining industry, and state-of-the-art machine learning methods for mining high-dimensional time series data. We describe an international data mining challenge organized to facilitate our study. We also demonstrate that the technique, which we employed to construct an ensemble of regression models (presented in Section 4.3) together with the sliding window feature extraction framework (Section 4.1.2) were able to outperform other approaches used by participants of the challenge. Finally, we explain how we utilized the data obtained during the competition for the purpose of research on the cold start problem in deploying decision support systems at new mining sites.

5.2.1 Seismic Hazards in Coal Mines

Coal mining is one of the most important branches of heavy industry in the world, with the employment level exceeding 3.5M people worldwide [179]. As briefly outlined in Section 5.1, there are many threats that may be encountered by miners working in underground coal mines. An important aspect of safe and efficient coal mining is the prediction of seismic hazards, particularly those related to high-energy destructive tremors, which may result in rock-bursts [48]. Safety refers to saving workers from accidents and injuries, while efficiency refers to unplanned shut-downs of longwall systems. From this perspective, proper prognosis of potentially dangerous methane concentrations [345] and seismic events [109] constitutes one of the most important challenges that should lead toward improving the safety and reducing the costs of underground coal mining.

More and more advanced seismic and seismoacoustic monitoring systems allow for a better understanding of rock mass processes [109] and defining seismic hazard

prediction methods [124]. Seismic hazard assessment and prediction methods, among others, include: probabilistic analysis [228, 272] that predicts the energy of future tremors or a linear prediction method [214], which can be used to predict aggregated seismic and seismoacoustic energy emitted from a mining longwall. Both methods perform analysis in a given time horizon. An application of data clustering techniques to seismic hazard assessment was presented in [232]. There are also approaches to the prediction of seismic tremors by means of artificial neural networks [195] and rule-based systems [197]. The accuracy of the methods created so far is, however, far from perfect. These methods often require a special, non-standard measuring apparatus and that is the main reason why some of them are not applied in mining practice.

Microseismic monitoring and multi-parameter indices may be also considered as a tool for the early warning of rock-bursts [96]. In the context of dealing with uncertainties in geomechanical underground works, particularly interesting are techniques that apply the Bayesian modeling approach [264]. Rule induction and decision tree construction techniques were also applied for this purpose [344]. There are also applications of machine learning methods to diagnostics of mining equipment and machinery [386]. The issue of mining devices diagnostics was raised among others in [265]. Still, expert systems are currently the most popular method of natural hazard prediction in the area of underground coal mining.

Two basic methods are routinely used by experts for the assessment of seismic hazards in Polish coal mines. These methods are often called *seismic* and *seismoacoustic*, respectively [197]. In Appendix B.1, we briefly describe both methods of seismic hazard assessment. The seismic and seismoacoustic methods are the result of the work of many domain experts and serve as a current standard in the Polish mining industry. Therefore, estimating the accuracy of those expert methods for natural hazard assessment and comparing their reliability with automatic prediction models constructed using statistical and machine learning techniques is of the utmost importance. This was one of the objectives of the presented research.

Processes related to the seismic activity are often considered the hardest types of natural hazards to predict. In this respect, they are comparable to earthquakes. Seismic activity in underground coal mines occurs in the case of a specific structure of geological deposits and due to the excavation of coal. Factors which influence the nature of seismic hazards are diverse. Relationships between those factors are very complex and still insufficiently recognized. To provide protection for people working underground, systems for active monitoring of coal extraction processes are typically used. One of their fundamental applications is to screen seismic activity in order to minimize the risk of severe mining incidents. Such a situation occurs in the Upper Silesian Coal Basin, where the additional conditions are related to the multi-seam structure of the coal deposit [168].

In almost all mines in this region, there are systems that detect and assess seismic activity degrees. The current industry standard in this regard (and the regulations imposed by Polish law) involves manual assessment of hazards by mining experts. However, the question remains whether the existing systems and expert methods take full advantage of the available data in order to provide their users with the maximum possible prediction accuracy. Moreover, it is important to design seismic

hazard prediction methods that can adapt to new conditions. There is also a question of whether the way in which currently deployed systems work is sufficiently clear and comprehensible, so the users can properly interpret their results and react in case of possible false emergencies.

Table 5.6: Basic characteristics for data obtained from different working sites. The first column shows working sites ids, whereas the subsequent ones present information regarding initial expert assessments of the working site's safety, the number of data samples in the training and test sets, and the percentage of cases with the *'warning'* decision label.

main working site ID	initial assessment	number of training cases	number of test cases	training warnings (percent)	test warnings (percent)
146	a	5591	98	0.0014	0.0000
149	b	4248	98	0.0718	0.0018
155	b	3839	98	0.1681	0.0094
171	a	0	49	0.0000	0.0000
264	b	20533	0	0.0039	0.0000
373	b	31236	0	0.0113	0.0000
437	b	11682	0	0.0041	0.0000
470	c	0	258	0.0000	0.0078
479	a	2488	35	0.0000	0.0000
490	a	0	160	0.0000	0.0500
508	a	0	58	0.0000	0.0172
541	b	6429	5	0.0087	0.0000
575	b	4891	253	0.0045	0.0012
583	b	3552	215	0.0021	0.0029
599	a	1196	363	0.0148	0.0289
607	b	2328	209	0.0000	0.0000
641	a	0	97	0.0000	0.0103
689	b	0	83	0.0000	0.1205
703	a	0	145	0.0000	0.0069
725	b	14777	330	0.0920	0.0021
765	a	4578	329	0.0000	0.0022
777	b	13437	330	0.0000	0.0009
793	b	2346	330	0.0000	0.0045
799	a	0	317	0.0000	0.0000
total	-	133151	3860	0.0226	0.0508

5.2.2 AAIA'16 Data Challenge

The complexity of seismic processes and the imbalanced distribution of the positive (e.g., *'warning'*) and negative (*'normal'*) examples is a serious difficulty in seismic hazard prediction. Commonly used statistical methods are still insufficient to achieve good sensitivity and specificity of the predictions. Therefore, it is essential to search for new and more efficient techniques of natural hazard assessment, including methods

derived from the field of machine learning. By organizing an international data challenge related to seismic hazards assessment as an open, on-line competition we aimed to conveniently review and evaluate the performance of the available state-of-the-art methods. Furthermore, this allowed us to verify not only the viability of the predictive models but also whole analytic processes, including preprocessing, feature extraction, model construction, and post-processing of predictions (e.g., ensemble approaches).

AAIA'16 Data Mining Challenge: Predicting Dangerous Seismic Events in Active Coal Mines took place between October 5, 2015, and February 27, 2016. It was organized at the KnowledgePit platform, under auspices of 11^{th} International Symposium on Advances in Artificial Intelligence and Applications (AAIA'16) which is a part of the FedCSIS conference series. The task in this competition was related to the assessment of safety conditions in underground coal mines with regard to seismic activity and early detection of seismic hazards.

The data set provided to participants was composed of readings from sensors that monitor the seismic activity perceived at longwalls of different coal mines and measure the energy released by so-called seismic bumps. Each case in the data was described by a series of hourly aggregated sensor readings from a 24 hour period. The provided data also contained information regarding the intensity of recent mining activities at the corresponding working site, coupled with the latest assessments of the safety conditions made by mining experts (for instance, ratings obtained using the seismic and seismoacoustic methods – described in Appendix B.1). To further enrich the available data, for each working site that occurs in the data set, some additional meta-data were made available, such as identifiers of the mine and region where the working site is located or a working site's height. The detailed list of all data attributes is available in Table A.2 in Appendix A.2.

Participants of the competition were asked to design a prediction model which would be capable of accurately detecting periods of increased seismic activity. In particular, the target attribute in the provided data (the decision) indicated cases for which the total energy of seismic bumps observed in a following 8 hour period exceeded the warning level of $5 \cdot 10^4$ Joules (i.e., the energy released in the period starting after the last hour of aggregated readings describing the case and ending 8 hours later). In total, the provided data was described by 541 main attributes and 6 additional features related to particular working sites. Most of the attributes were numeric, but there were also a few symbolic (qualitative) ones, e.g., assessments made by experts. The competition's data correspond to over 5 years of readings which, to the best of our knowledge, makes this research the most comprehensive study related to this domain, conducted anywhere in the world.

The data set was divided into a training part, which was made available to participants along with the corresponding decision labels, and a test part. The labels for the test set were hidden from participants. The division of cases between the training and test sets was made based on time stamps. In particular, the training data set corresponding to a period between May 5, 2010 and March 6, 2014. It consisted of 133151 data rows, each corresponding to a different 24-hour period, overlapping for consecutive cases. The test data covered the period between March 7, 2014 and June 24, 2015. Unlike the training set, to facilitate the objective evaluation

Table 5.7: Final results and number of submissions from the selected, top ranked teams in the AAIA'16 Data Challenge. The last row shows results obtained solely from assessments made by mining experts that were available in the data (see Appendix B.1).

Method	No. of submissions	AUC
Grzegorowski M. [138]	2	0.9396
Milczek et al. (Deepsense Inc.) [262]	111+31	0.9393
Tabandeh Y. (Golgohar Inc.[1])	54	0.9342
Podlodowski [298]	71	0.9336
Kurach & Pawłowski [221]	32	0.9312
Başak et al. [150]	30	0.9297
...	...	...
experts assessment (18^{th} place)	–	0.9196

of solutions and to prevent a common problem with so-called data leakage [204], the test cases were not overlapping and provided in random order. For this reason, the test set used in the challenge was much smaller than the training data but still covered a period of nearly 16 months.

Table 5.6 shows some basic data characteristics from each working site that was used in the competition. It is worth noting that not all working sites present in the training data also appeared in the test set and there were a few working sites that were present in the test data but not in the training set. Such a situation reflects a real-life problem when the exploration of coal shifts to a new site for which there is no data available. A similar issue can also be identified within other domains, e.g., recommender systems, and is commonly referred to as the *cold start* problem [364]. A fact worth noticing is also that the distribution of cases with a *'warning'* decision label is quite uneven for different working sites.

Solutions submitted by participants had a form of scores assigned to the test cases (i.e., real numbers, which could be interpreted as likelihoods of 'warning' signals). In practical applications related to the monitoring of safety conditions, such a form of predictions is more valuable than the exact decision labels because it allows for tuning the sensitivity of the utilized model. Due to imbalanced distributions of decision labels, the quality of each submission was measured using Area Under the ROC (AUC). The AUC measure explicitly relates the true alarm rate to the false alarm rate and, thus, is appropriate for measuring the performance of prediction models in a situation when underestimating the risk of a minority binary class (i.e., a seismic event) is significantly worse (in our case in terms of safety) than overestimating the risk.

Among the registered teams, 106 were active, i.e., submitted at least one solution. Table 5.7 shows scores achieved by the selected, top-ranked teams. In total, we received 3236 solutions, of which 3135 were correctly formatted and successfully passed the evaluation procedure. In Section 5.2.4, we explain how we used those submissions in our post-competition analysis of the cold start problem in the deployment of predictive models for new working sites. Additionally, 50 of the participating teams provided reports describing their approach, e.g., [138, 262, 421]. These reports turned out to be a valuable source of knowledge regarding the

state-of-the-art solutions in the predictive analysis of time series data related to early detection of seismic hazards.

5.2.3 Construction of a Seismic Hazard Assessment Model

In this section, we focus not only on the constructed prediction models but also on data processing stages that were designed to let it work within a big data processing environment, and particularly with multi-sensor data streams. The highest result in the final evaluation of AAIA'16 Data Challenge was obtained out-of-the-box by the solution trained with the framework described in Section 4.3. The Grzegorowski M. [138] method (Table 5.7)) is based on the framework presented in Chapter 4 – and had been successfully applied to a similar problem, namely, the prediction of dangerous methane concentration levels in corridors of coal mines (Section 5.1). The fact that we were able to reuse this approach confirms its attractive generality. The overall work we spend on the framework configuration, data preprocessing, feature extraction and selection, models training, and ensemble blending for the purpose of seismic data prediction did not exceed 2 hours.

Let us now take a closer look at the best performing solution – Grzegorowski M. [138] (see Table 5.7) – of the AAIA'16 Data Challenge. In order to provide high quality assessments, this solution constructed an ensemble of diverse logistic regression models (Section 4.3). The diversity is obtained by employing a variety of models computed on various subsets of attributes and examples (Section 4.1.2). By aggregating predictions of those models using the Algorithm 9, we were able to obtain robust performance even for new mining sites. As a result, the final ensemble minimized the impact of a concept drift [85, 95] and achieved a better quality and robustness of prediction than models used by all other teams participating in the competition.

A scheme of the whole feature extraction process is the same as depicted in Figure 4.5. In the *'Map'* phase (compare steps 2 and 3 in Figure 4.5), each data row was divided into sub-series of numerical values from various sensors, a set of static and aggregated features and, in the case of the training set, also a label. The labels, as well as the static attributes from experts, were transferred to the *'Reduce'* phase unchanged while the time series were subjected to the feature extraction process described in Section 4.1.2. In the *'Reduce'* phase (steps: 4 and 5 in Figure 4.5), all the attributes obtained for each row were combined again.

The design of our model and particularly the method for construction of an ensemble was largely affected by the imbalanced distribution of *'warning'* cases in the data (only about 2.3% of all objects). Firstly, we drew a number of random samples that contained between 10000 and 20000 objects from the training set. The samples differed in the number of objects from the *'warning'* class – it was assured that each contained a minimum of 1000 and a maximum of 2000 such cases. Objects within a particular sample were unique, but they could be repeated between different samples. Such prepared samples supported the ensemble feature selection technique to yield more robust results [329]. All steps of the feature selection, models training, and the construction of the final ensemble are presented in Figures 4.6 and 4.7.

It is worth noticing that the selection of attribute subsets was carried out using

a technique that originated within the theory of rough sets (cf. Section 2.5). Similarly, as in the FE+DAAR (Table 5.2) approach presented in Section 5.1.3, the DAAR heuristic for computation of approximate decision reducts was applied to find relatively small subsets of relevant features. Furthermore, we decided to combine a few approximate reducts into a single attribute subset to extend the feature space for each logistic regression model while increasing their diversity (compare steps 8 – 10 in Figure 4.6) – the combined reducts were randomly matched. Only significantly different subsets were maintained for the purpose of model training, the rest were filtered out.

In the next steps (compare Figure 4.7), the obtained subsets of attributes and objects with a more balanced distribution of classes (see, e.g., SMOTE oversampling technique for sensor readings and ensemble learning [237]) were used to train logistic regression models using pre-selected algorithms. The most important models were used to form an ensemble. Criteria for selecting models for the ensemble considered a quality of individual regressors as well as the degree of diversity of a resulting collection of models. The course of the experiment is presented in Algorithm 9 and in Figure 4.7. All the processing steps were implemented in R environment for statistical computing using additional libraries, e.g., *rpart*, *e1071* and *RoughSets* [319].

The final solution was an ensemble of diverse logistic regression models which interpret the *'warning'* label as 1 and the *'normal'* label as 0. The diversity was achieved by training the models on different subsets of attributes and objects. Algorithm 9 guarantees that a model can be included only if it is accurate enough on validation data and sufficiently different from already selected predictors (in that case, correlation of its prediction with predictions of other models is small enough). This could be seen as a method for increasing the robustness of predictions in the case of noisy and heterogeneous data.

The final ensemble consisted of 8 different regression models which were calculated using three various algorithms, namely: regression trees (calculated using the implementation from the *rpart* library), SVM regression, and a generalized linear model. These particular models were selected:

- five regression trees calculated with rpart (default settings),
- two SVM models with different kernel functions:
  - SVM_1 - linear kernel, cost: 1, eps: 0.1
    (the number of support vectors: 2968),
  - SVM_2 - radial kernel, cost: 1, gamma: 0.07143, eps: 0.1
    (the number of support vectors: 7171),
- one logistic regression model computed with glm.

A comparison of the selected solutions to predictions that were based solely on assessments made by experts revealed that more complex models were able to quickly attain significantly higher scores for working sites with available training data. In the case of the remaining working sites, the advantage of complex prediction models was not that clear. The average results for selected models in phase 6 were only

slightly higher, however, for a part of the investigated solutions, the difference was much more favorable than for others.

Table 5.7 shows scores achieved by the selected, top-ranked teams. The second highest result, achieved by Milczek et al. solution [262], was based on a mixture of multiple gradient-boosted trees [64, 114], extremely randomized trees [34, 122] multi-task logistic regression and linear discriminant analysis models [273]. Interestingly, when we computed the Spearman's correlation between predictions of our model and the Milczek et al. (Deepsense Inc.) [262] model, it turned out to be relatively small (≈ 0.77). When we combined these predictions by averaging ranks of predicted values for the test cases, we obtained a higher AUC than those of the individual models (0.9421 vs. 0.9396 and 0.9393). This result highlights the benefit of using diverse prediction models for constructing ensembles.

In general, an overview of the most successful approaches in the competition suggests that the key steps to achieve a good result in this task included:

1. Extracting relevant features (computing a new data representation) that aggregate time series data and are robust with regard to a concept drift.

2. Designing an appropriate evaluation procedure for testing performance of used prediction models and tuning their parameters.

3. Using ensemble learning techniques for blending predictions of simpler models.

Such a general approach, which is strongly dependent on feature engineering, was employed by eight out of the ten top-ranked teams. A slightly different methods utilized deep neural networks (DNN) to automatically learn a representation of data [414]. Some details regarding DNN approaches may be also found in [221].

It is here very important to stress out that a solution based solely on domain experts' assessments achieved the 18^{th} position. It confirmed that data mining techniques may outperform experts in forecasting seismic hazards. It also showed that this problem is not easy to solve even using state-of-the-art ML methods. Furthermore, a closer analysis of the reports submitted by the most successful teams revealed that the attributes corresponding to experts' assessments were commonly used by their models. Interestingly, all submitted solutions performed significantly better when the set of input data included the most recent evaluations provided by experts. It clearly shows how important the domain knowledge is for the efficient assessment of seismic hazards in coal mines. Furthermore, as we show in the subsequent section, expert knowledge allowed us to successfully transfer the knowledge to the new working sites.

5.2.4 The Cold Start Problem

The cold start problem is an important practical issue that is related to real-world applications of many decision support systems. In the case of coal mining, it typically appears when a system for monitoring natural hazards is being deployed for new, previously unexplored longwalls. One of the research objectives, motivating the organization of AAIA'16 Data Mining Challenge (described in Section 5.2.2), was to

investigate the severity of this problem in the context of systems for early detection of periods of increased seismic activity.

To gather comprehensive data about the impact of the size of available data on the quality of predictions for a given working site, the training set in the competition was divided into five separate parts and the challenge was split into six phases. Table 5.8 shows some basic statistics related to the consecutive phases, including the maximum size of the data set available to participants in each phase of the challenge. After the start of the competition, only the first part of the training data was revealed. The four consecutive parts were made available in monthly intervals. In the sixth phase, which lasted for the last two weeks of the competition, all training data parts were revealed to all participating teams. Since in each phase a new subset of training data was made available to participants, we were able to verify the impact of this additional information by examining the quality of solutions submitted in consecutive phases.

Table 5.8: Basic statistics for each phase of the challenge, including: training set size, number of submissions (cases) as well as best, mean and standard deviation of AUC scores.

phase	training set size	no. of uploads	best score	mean score	std. dev.
1	79893	99	0.9290	0.8251	0.0672
2	93211	278	0.9320	0.8851	0.0587
3	106527	1377	0.9405	0.8307	0.1058
4	119839	363	0.9375	0.8772	0.0831
5	133151	513	0.9379	0.8857	0.0625
6	133151	505	0.9439	0.8942	0.0696

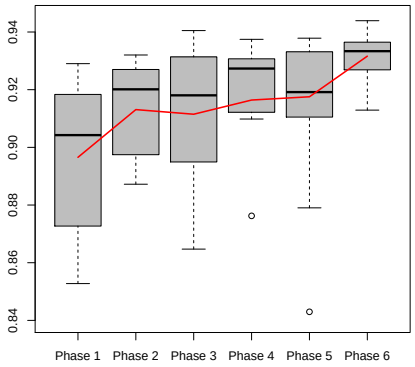


Figure 5.5: Distribution of the best AUC scores per team and phase.

A detailed analysis of the distribution of scores in time reveals some interesting observations. For the analysis, we only used valid solutions with a reasonable quality (we disregarded 'random' submissions and those which obtained a preliminary score lower than 0.65). Table 5.8 shows the mean and standard deviation of evaluation scores for each of the competition phases.

As an interesting observation related to the analysis of the results shown in Table 5.8, we may point that, starting at some point in time, the use of additional training data has a diminishing impact on the performance of prediction models. For instance, if we compare the average results from the second phase with the results from the fourth or fifth phase, we see that the difference is minimal. Even though in these phases we received a comparable number of submissions, and the available training set in the phase 5 was larger than in the phase 2 by nearly 43%. This was even less expected due to the fact that the data available in the phase 2 contained information about only 9 out of 21 main working sites present in the test data (these sites corresponded to $\approx 45\%$ of observations in the test set), whereas in the phase 5 this number was much higher (13 out of 21 sites; $\approx 70\%$ of observations). This suggests that models relatively quickly became saturated with the training examples.

To further confirm this observation, we analyzed the best solutions in each phase taking into account only the submissions from well-performing teams – that obtained AUC scores higher than 0.85 – results of such teams better reflect the performance of

the state-of-the-art models. Figure 5.5 visualizes basic statistics (min, max, quantiles, and the mean values) of the best AUC scores of those submissions. The average scores slightly increase from phase to phase. However, when we checked the statistical significance of the changes, it turned out that a significant difference (p-value lower than 0.01) occurred only between results from the fifth and sixth phases. For other consecutive phases, the p-values of the Wilcoxon test were always higher than 0.175.

Let us now thoroughly investigate the performance of top-ranked solutions submitted in each phase, with regard to individual working sites. For this purpose, we disregarded working sites for which there were no examples with the *'warning'* label in the test set. The reason for that was the inability to compute values of AUC on such data subsets. This way, for the remaining part of our analysis there were 15 working sites left, which corresponded to $\approx 81.5\%$ of observations in the test data. From solutions submitted in each competition phase, we chose 6 with scores in the top 10% for a given phase. Table 5.9 shows their average AUC values with respect to individual working sites. Additionally, the last two rows of the table give average values of AUC for working sites that were present in the training set and for those which were unavailable in the training data, respectively. Finally, the last column of Table 5.9 shows AUC values obtained for individual working sites using only the assessments made by experts.

For most of the working sites there is a statistically significant improvement (tested using t-test with a confidence level of 0.95) of results from the later competition phases in comparison to the first phase. However, in nearly all cases the improvement between the second and later phases becomes marginal (one exception is the working site with ID:599). Interestingly, there are working sites (e.g., ID:689, ID:777) for which there is a noticeable drop in the average quality of solutions between the second phase and phases 3, 4 and 5. Our partial explanation of this phenomenon is that in the case of site 689, there is no training data for this particular site, while in the test set, it is characterized by the highest percentage of 'warning' cases (over 12%). In the case of site ID:777, the situation is the opposite. For this site, there were many examples in the training data and their number was increasing in consecutive phases. However, all examples in the training set belonged to 'normal' decision class, whereas the test set contained a few observations from the 'warning' class (see Table 5.6). Such a distribution of labels could trick the models into thinking that all cases from this mining site should be 'normal', and as a result, decreased their performance. Interesting is also the fact that the top solutions obtained consistently higher scores for working sites that were not present in the training data.

The above observations show that having a sufficiently large data set, it is possible to construct efficient prediction models for the assessment of seismic hazards. The created models can outperform the currently used expert methods even for completely new working sites, as long as these sites have comparable geophysical properties and the same methodology is used for collecting new data. At this point, it is worth emphasizing that all the evaluated models were trained on both sensory data and domain experts' assessments. Such an approach allowed to significantly improve the quality of machine learning models since it encapsulated part of the knowledge about the particular conditions of each working site not covered by sensors. This is an important argument for considering interactive feature extraction processes and

Table 5.9: Average scores of top solutions for individual working sites, in different phases of the competition. Evaluations of expert assessments are given for comparison in the last column. Additionally, the last two rows display aggregated values (averages) for working sites with some data in the training set and the working sites without any available training data.

working site ID	phase 1	phase 2	phase 3	phase 4	phase 5	phase 6	expert's assessment
149	0.8984	0.9056	0.8523	0.9062	0.8766	0.9005	0.9306
155	0.6578	0.7328	0.7492	0.7393	0.7242	0.7487	0.6845
470	0.9749	0.9922	0.9876	0.9935	0.9922	0.9964	0.9707
490	0.8013	0.8122	0.8340	0.8021	0.7892	0.8289	0.8109
508	0.9825	0.9971	1.0000	0.9942	0.9854	1.0000	1.0000
575	0.9348	0.9845	0.9859	0.9826	0.9820	0.9825	0.9723
583	0.9000	0.9419	0.9363	0.9388	0.9370	0.9401	0.9280
599	0.8391	0.8585	0.8678	0.8445	0.8670	0.8710	0.8020
641	0.9809	0.9983	1.0000	0.9965	1.0000	1.0000	1.0000
689	0.7723	0.8812	0.8523	0.8685	0.8582	0.8938	0.8884
703	0.9346	0.9792	0.9826	0.9699	0.9873	0.9722	0.9722
725	0.8968	0.9188	0.9251	0.9151	0.9176	0.9099	0.8955
765	0.7989	0.7911	0.7367	0.7608	0.7423	0.7808	0.7587
777	0.9118	0.9354	0.9242	0.9252	0.9175	0.9408	0.9444
793	0.9499	0.9545	0.9585	0.9538	0.9361	0.9468	0.8868
avail. in training	0.8653	0.8915	0.8818	0.8852	0.8778	0.8912	0.8670
unavail. in training	0.9077	0.9433	0.9428	0.9374	0.9354	0.9486	0.9404

built-in human-computer interaction into machine learning processes.

5.3 Tagging Firefighter Posture and Activities

A fire ground is considered to be one of the most challenging decision-making environments. In dynamically changing situations, such as those occurring at a fire scene, all decisions need to be taken in a very short time. Since wrong decisions might have severe consequences, a commander of the response team is forced to act under huge pressure [180]. Based on several thousands of carefully analyzed reports, experts identified the "lack of situational awareness" as the main factor associated with major accidents among firefighters [133]. According to studies on causes of mortal accidents during actions of firefighters conducted by the Department of Homeland Security of the United States [5] over 43% of deaths at a fire scene was caused by stress or overexertion. Therefore, another critical way of increasing firefighter safety is by monitoring their kinematics and psycho-physical condition during the course of fire & rescue actions.

The computer systems for human activity recognition may help to reduce unsafe events, improving communication, and increasing the efficacy of incident

management. Human activity recognition using Body Sensor Networks (BSN) is a non-invasive system that is able to deliver information about person motion patterns, posture and specific actions performed [152, 227]. A network of sensors located on a firefighter body are used to gather kinematic (motion) data from different parts of the body that may be additionally complemented with physiological data sensors for vital function monitoring. Afterward, sensor readings are transferred, pre-processed and various machine learning classification techniques may be applied in order to estimate the current activity. In this section, we present a practical application of the presented framework for interactive feature extraction in the fire and rescue domain that refers to BSN data analysis.

5.3.1 Additional Constraints and Requirements

The aim of our research was to assess how the automatic feature extraction and classifiers learning (without parameters tuning) can cope with the multi-target learning problem [19, 426]. The evaluated mechanism was previously applied for the purpose of processing multiple streams of readings generated by sensor networks in coal mines (Sections 5.1 and 5.2). Hence, one of our objectives was to assess the versatility of the developed framework across significantly different domains of application. Working on the solution, we imposed a few additional constraints and requirements that are essential for the emergency and threats detection domains:

1. The overall time spent on solving the problem must not exceed a total of 2MD (two man days - that is 16 h).

2. The overall computation time required to prepare data and train the classifiers must not exceed the total of 10 minutes.

3. The total time required to pre-process a single row of data to a format accepted by a classifier and assignment of both labels must not exceed one second.

The first of the imposed restrictions was intended to evaluate the possibility to adapt quickly to a new domain with a satisfactory quality of the model. We put a requirement that the time for configuring the framework, processing data, training, and applying the model should not exceed 2 MD which has been recognized as sufficient for researchers to become familiar with the task and to adjust original data representation to formats accepted by the evaluated feature extraction mechanisms.

The second point posed a constraint on the time necessary to re-train the model on new data. After a certain time, the quality may fall below the predetermined threshold due to, e.g., concept shift or drift [128, 240]. We assumed that the time required for re-training the model should not exceed 10 minutes.

The last point, we consider as the most important because it imposed limits on the permissible delay in the operation of the pre-processor and classifiers when acting in a production environment. According to the assumptions, maximum delay between data collection, complete processing, and labeling of a single row of data should not exceed one second. This was one of the main reasons for excluding from consideration all object-based methods as well as heavy classifier ensembles.

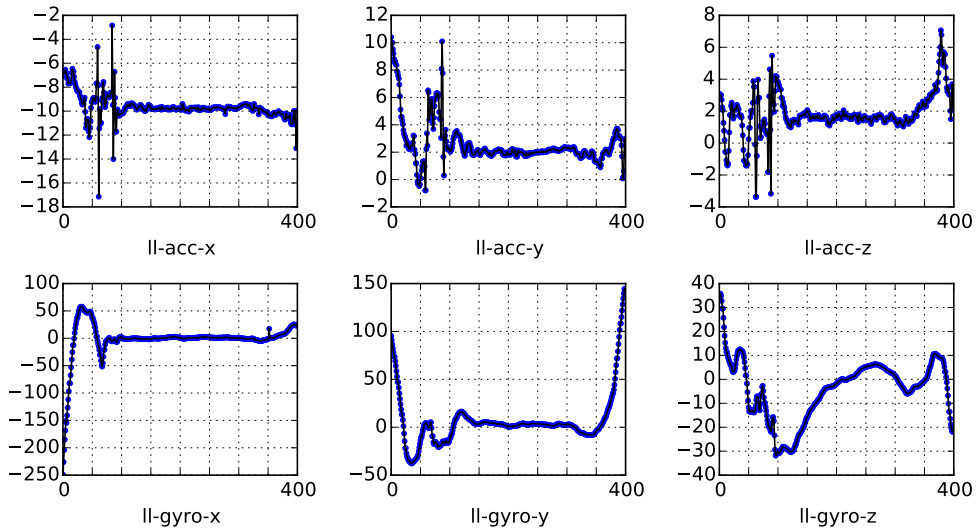


Figure 5.6: Time series form an accelerometer (measured in m/s^2) and a gyroscope (measured in deg/s).

5.3.2 AAIA'15 Data Challenge

The competition: Tagging Firefighter Activities at a Fire Scene [260] – organized within the frame of the International Symposium on Advances in Artificial Intelligence and Applications[2] – concerned the problem of an automatic assignment of labels (activities) to short series of readings from sensors that monitor activities and movements of firefighters during an action. The aim of the competition was to maximize a balanced accuracy measure which is defined as an average accuracy within all decision classes. It was computed separately for the labels describing the posture and main activities of firefighters. The final score is a weighted average of balanced accuracies computed for those two sets of labels and is defined as follows:

$$score(s) = \frac{BAC_p(s) + 2 \cdot BAC_a(s)}{3} \tag{5.2}$$

Where BAC_p is the balanced accuracy for labels describing the posture and BAC_a for the main activity. Recall the definition of the balanced accuracy (BAC):

$$BAC(preds, labels) = \frac{\sum_{1<i<l} ACC_i(preds, labels)}{l}$$

$$ACC_i(preds, labels) = \frac{\mid j : preds_j = labels_j = i \mid}{\mid j : labels_j = i \mid}$$

The data provided in the competition were obtained during training exercises conducted by a group of eight firefighters from the Main School of Fire Service. The sensors placed on a chest were registering vital functions, while the sensors placed on the torso, hands, arms, and legs were registering movements of a firefighter. Along

[2] AAIA'15, https://fedcsis.org/aaia

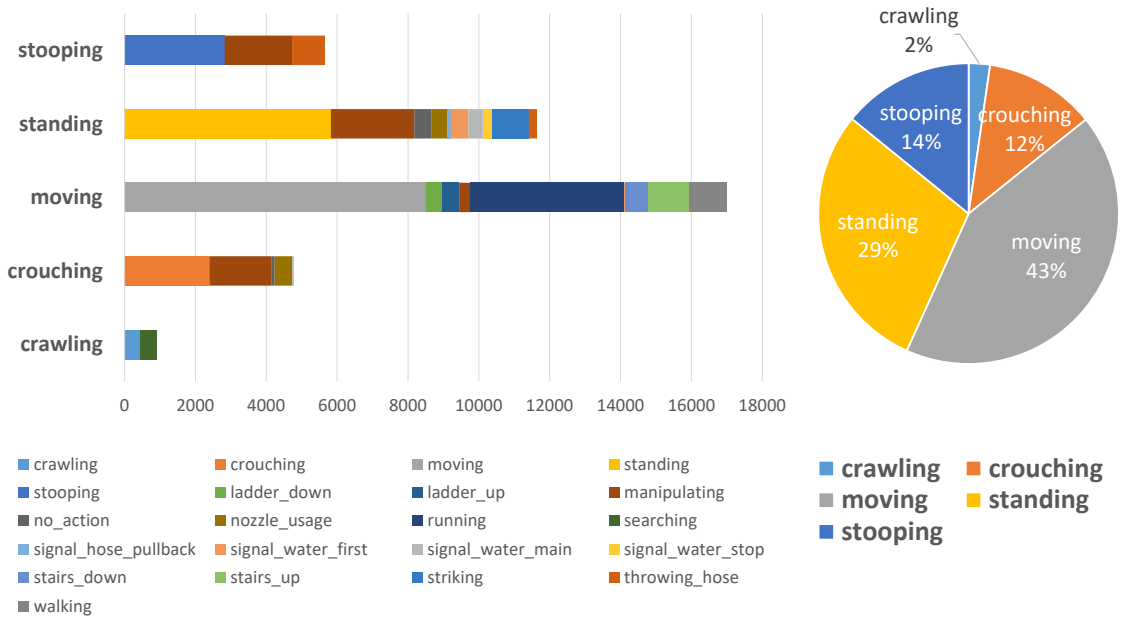


Figure 5.7: Inter-dependencies between posture and activity labels in training data.

with recording the data from sensors, all training sessions were also filmed. The video recordings, firstly synchronized with the sensor readings, were presented to experts who manually labeled them with actions performed during the exercises. The training and test data sets contain 20000 rows and 17242 columns each. The data are available online on the KnowledgePit platform as CSV files. The considered task was even more challenging since the training and test data sets consist of recordings from disjoint groups of firefighters.

Each single row in data sets corresponds to several short time series with length equal to approximately 1.8 s. The first 42 columns contain basic statistics (like mean, standard deviation, maximum, minimum, etc.) of data from sensors monitoring a firefighter's vital functions over the given, fixed time period. The raw readings for the vital functions were recorded using Equivital Single Subject Kit (EQ-02-KIT-SU-4) fitted with two medical-quality ECG units, heart rate and breath rate units, and thermometers for measuring skin temperature. The remaining columns contain readings from a set of kinetic sensors that were attached to seven places on a body, i.e., left leg, right leg, left hand, right hand, left arm, right arm, and torso. The enumerated body areas correspond to the following name prefixes in data: $ll, rl, lh, rh, la, ra, torso$, respectively. An infix – *acc* or *gyro* – refers to an accelerometer (dynamic bandwidth: +/- 16G) or gyroscope (scale up to 2000 deg/s), respectively. Each sensor of both types produced three readings corresponding to the three dimensions. A suffix x, y, or z indicates the axis readings came from. An average time difference between consecutive sensory readings in the data is 4.5 ms. Eventually, time series are divided into 400 chunks that represent consecutive points in time. Figure 5.6 contains exemplary, six time series (each consists of 400 values that correspond to approximately 1.8 s) from a set of sensors placed on a left hand of a firefighter performing an exercise.

The above description shows the details of the values arrangement in the data.

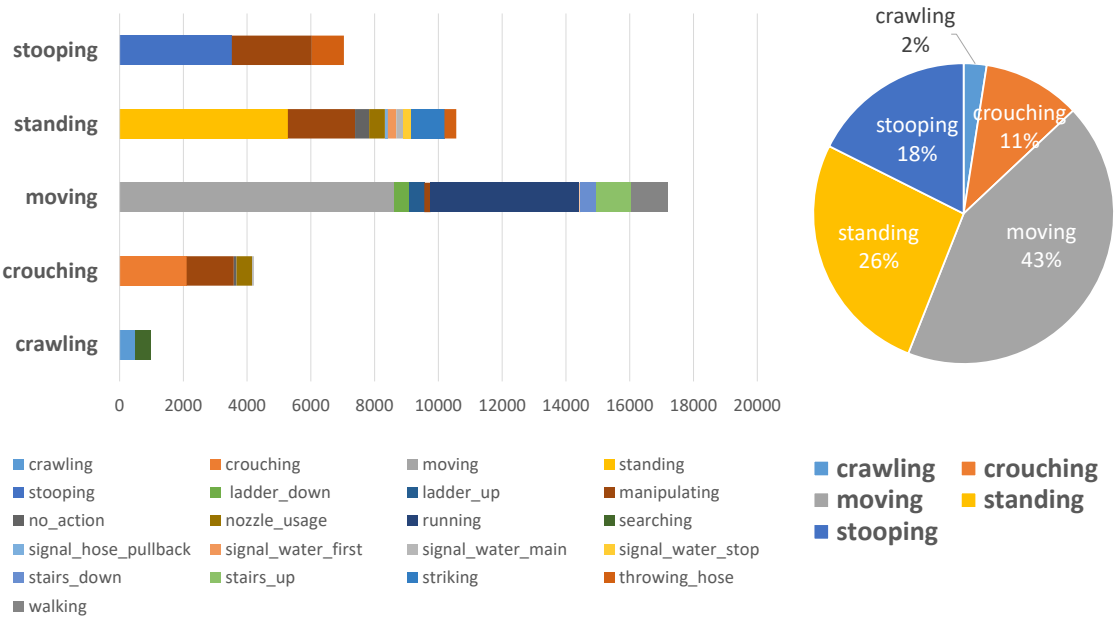


Figure 5.8: Inter-dependencies between posture and activity labels in test data set.

In the frame of the experiment, we considered each row as a separate data set containing readings from many sensors. Values from the vital sensors were aggregated externally but the kinetic ones were provided in the raw form of time series - this setup was out-of-the-box covered by the framework - compare Figure 4.5 in Section 4.3. However, the preliminary data analysis revealed two interesting characteristics of the investigated data. The posture and activity labels were not independent [203] – see Figures 5.7 and 5.7). Furthermore, the analysis revealed an imbalance in label distribution [392, 404].

5.3.3 Feature Extraction

For the purpose of posture and activity recognition, we processed the data with three configurations of a sliding window mechanism (Section 4.1.2). As presented in Figure 5.9, every time series were split into 1, 2, and 5 consecutive, non-overlapping sliding windows, respectively. If there were more than one window generated for the time series we extracted so-called inter window statistics (in addition to those included in a basic window) – that is a set of features expressing changes of attribute's values between a pair of consecutive windows (Section 4.1.2).

According to the task description, the kinetic sensors (accelerometers and gyroscopes) used during the exercises have symmetric scales with 0 as their neutral reading. The specificity of the firefighter activities like walking, running, moving up the stairs or ladder, could cause the readings to be more significant when considered as a group – e.g., a whole tuple (x, y, z) from a given accelerometer rather than separate readings x, y, and z. For that reason, we introduced a concept of so-called *virtual sensors*. Besides applying the aggregate functions to the original time series available in the delivered files, we implemented an idea of creating artificial time series derived from the original ones. The virtual sensors were created on the basis of one or more

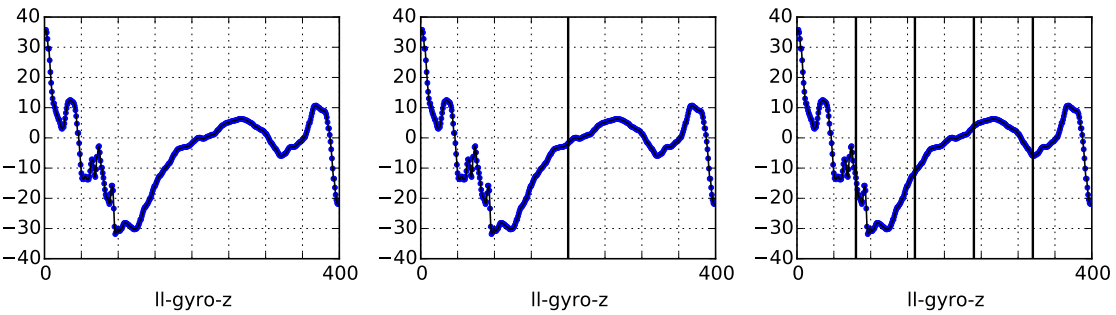


Figure 5.9: Illustration of the sliding window configurations. Time series were processed with varied granularity, ranging from the statistics computed for the whole time series, to calculate them for 2 or 5 shorter, non-overlapping sliding windows which divided the time series to the parts of equal length.

time series from other sensors (whether original or virtual) by applying a particular function. In our solution, we decided to create virtual sensors for readings from all accelerometers and gyroscopes' axes separately, applying an *abs* (absolute value) function. We created also virtual sensors for readings grouped in tuples (x, y, z) for each kinetic sensor – computing the Manhattan and Euclidean norms for the (x, y, z) vectors. An example that illustrates the concept of virtual sensors used in our solution is presented in Figure 5.10.

Along with so far mentioned attributes, we additionally extracted more domain-specific features, e.g., a sum of the selected features for the left and right hand or a sum for the left and right leg – to exclude the symmetry of right- and left-handed people. This was important because the training samples were created based on the behavior of different people than the test samples. Moreover, training and test set data were acquired during observation of a small group of firefighters, hence the training sample could not contain all possible patterns. All extracted statistics were joined together, in a sense of appending all their values in a data table, and served as an input for the further steps of data analysis and experiments (see the subsequent steps – 5 and above – in Figure 4.6 and in Figure 4.7).

The above-mentioned adjustments and re-configurations of the presented FE framework were performed in stages in an interactive way. In Figure 5.11, the ultimate schema of feature extraction process is presented. The extracted data sets had a total of 27177 attributes, due to 3 different sliding windows configurations: 2199 – one sliding window per time series; 6315 – two sliding windows per time series, and 18663 – five sliding windows. All identifiers and all constant attributes were removed from data. In the process of feature selection, we employed a wrapper approach. In the forward propagation phase, we had been progressively enlarging the number of candidate features and making periodic evaluations with SVM model after each step. Ultimately, 163 most significant attributes were selected for the purpose of model training.

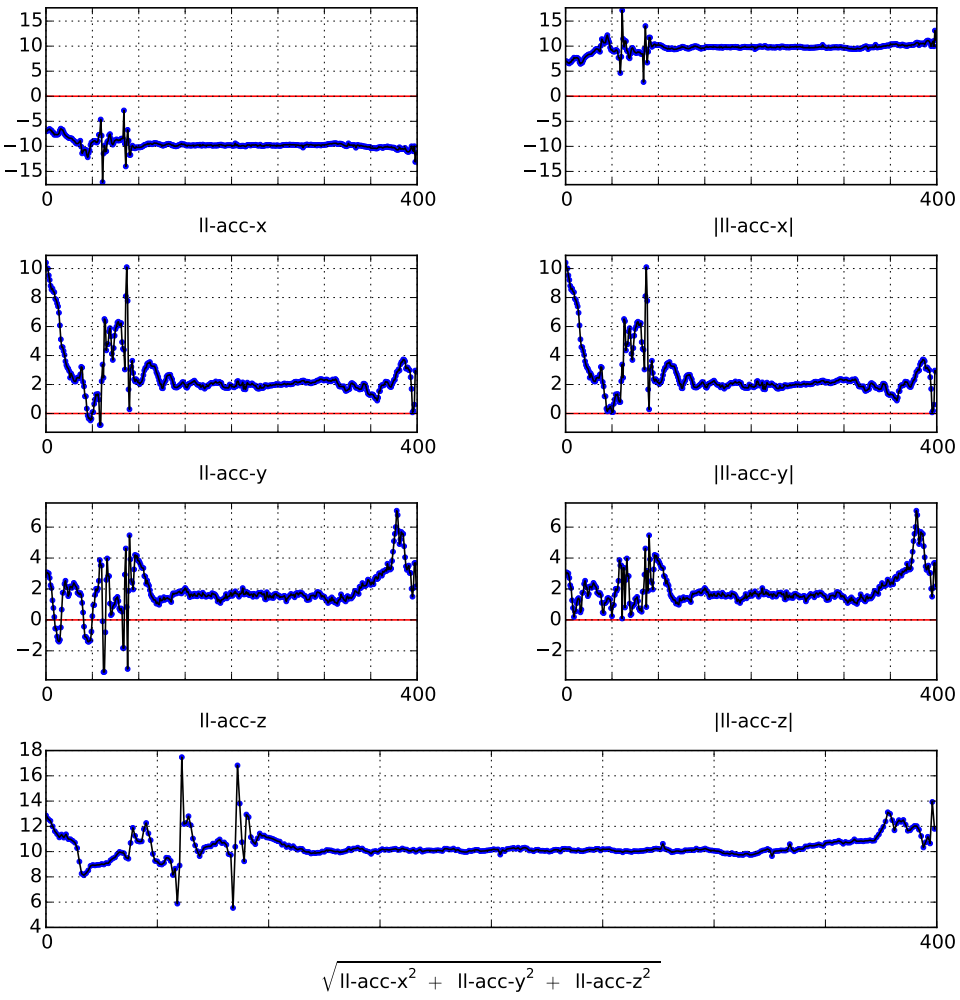


Figure 5.10: An example of virtual sensors extracted by applying an absolute value function and the Euclidean norm to the original time series.

5.3.4 Model Training

The main multi-target learning problem of labeling multivariate time series with many labels that are interdependent (Figures 5.7 and 5.7) may be modeled and solved in a number of ways [19, 426]. One of the options could be transformation to a typical classification problem by training classifiers to solve posture and activity recognition independently. Another option could be creation of new labels corresponding to couples of posture and activity, e.g., for posture: "moving" and action "stairs_up" the combined label would be "moving_stairs_up". Such an approach would incorporate the additional knowledge about dependencies between labels, on the other hand, the number of cases for niche activities would be relatively small. The way in which the assessment of the solutions is defined (eq. 5.3.2), that is uneven importance of labels for posture and activity, encourages to consider various concepts like a multi-label

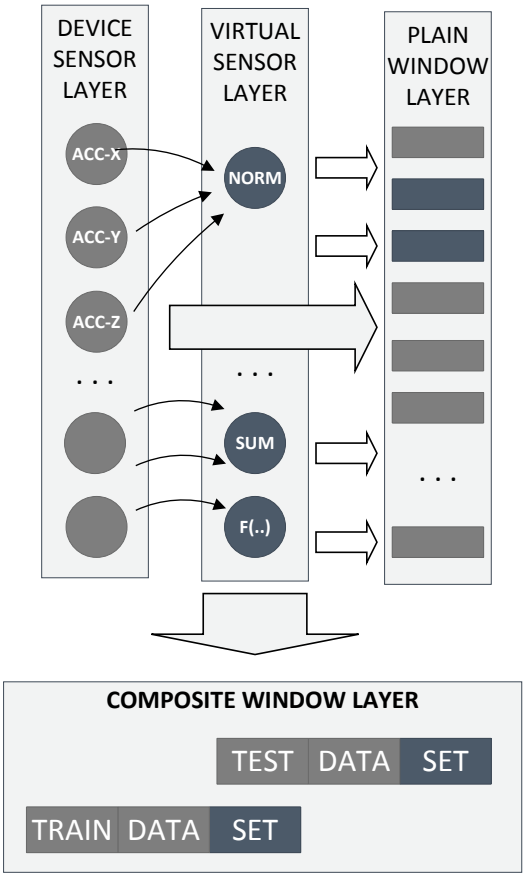


Figure 5.11: Pre-processing and feature extraction.

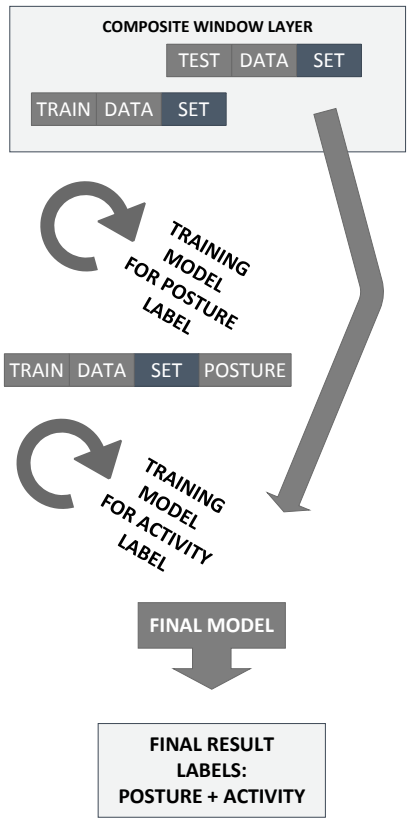


Figure 5.12: Classifier chain.

classification with label ranking [116] or a graded multilabel classification [65]. The experimentation with label power-set methods [317, 408], however, did not provide satisfactory results.

Ultimately, we decided to follow ensembles of classifier chains (ECC) [316]. This approach involves linking together classifiers in a chain structure [405], such that posture label predictions become features for activity classifiers. Class imbalance is an intrinsic characteristic of analyzed multi-label data (Compare Figures 5.7 and 5.7). Some of the labels in data were associated with a small number of training examples. In general, class imbalance poses a key challenge that plagues most multi-label learning methods [241]. Classifier Chains [316] – one of the most prominent multi-label learning methods – is no exception to this rule, as each of the models it builds is trained on all positive and negative examples of each label. To make a ECC resilient to class imbalance, we coupled it with over- and under-sampling (recall STEP 6 in Figure 4.6) [363].

Experiments were implemented and carried out in the R software environment with additional packages. We experimented with decision trees, random forest, and SVM models [203]. To align with agreed constraints, that is: not to exceed 10 minutes of model training and at most 1 second for single input processing and labeling, the final solution was based on two SVM models with radial kernel set up in a classification chain. In Figure 5.12, the schema of classifier chain training – carried

out in order to solve the problem of labeling sensor time series with posture and main activity of a firefighter – is presented. A model responsible for recognizing a firefighter's posture was trained on the 163 attributes. The (second in the chain) SVM model which classified the data with a main activity had one additional attribute – the prediction for posture label. Although constrained effort in solving the problem (limited with a maximum of 16 hours), the final evaluation reached $BAC = 0.72$ and significantly exceeded the organizer's baseline $BAC = 0.6$.

In the final embodiment, the total time of training classifiers did not exceed 7 minutes. Extraction of all the features, including those for both: raw and virtual-sensors readings, took approximately 450 milliseconds per a single csv file row. The post-processing, including assignment of the labels, was performed in R software environment for statistical computing and consisted of: importing data 1.5 millisecond per record (overall 30 seconds per 20000 rows of test data set), feature selection 0.5 millisecond per record (overall 10 seconds per 20000 rows of test set) and labeling 3.5 millisecond per record (classification with SVM of both labels for 20000 rows took in total 70 seconds).

5.4 Spot Instances Price Prediction

The ability to analyze the available data is a valuable asset for any successful business, especially when the analysis yields meaningful knowledge. Analytical data processing has become the cornerstone of today's businesses success, and it is facilitated by Big Data platforms that offer virtually limitless scalability. The storage technologies with high level of compression that support stream data collection and analytics [21,354] as well as the data processing and integration tools [51,320,420] , which can scale up to thousands of compute resources [87,136,172] allowed companies to store and analyze data collected from ubiquitous sensors. Furthermore, cloud computing has emerged as an important paradigm offering a variety of low-cost hardware and software in pay-as-you-go pricing model [399], which is particularly convenient for Big Data analytics [339].

Cloud computing offers a number of Big Data solutions related to scalable storage, processing, and sophisticated business analytics. Due to the growth of Big Data over cloud, cost-effective allocation of appropriate resources has become a significant research problem [205, 339]. Minimizing the total cost of ownership (TCO) for the infrastructure supporting Big Data is considered a very challenging task. The number of available pricing models on the cloud markets is overwhelming, but it is worth paying special attention to two of them, in particular: the on-demand and spot markets. The first one represents the pay-as-you-go cloud model, and today is the most common way the resources are provisioned. The second one allows customers to save up to 90% of costs by using the cloud data centers' idle servers.

In this section, we show that, by analyzing spot instance price history and using ARIMA models, it is feasible to leverage the discounted prices of the cloud spot market [145]. In particular, we evaluate savings opportunities when using Amazon EC2 spot instances comparing to on-demand resources. The performed experiments confirmed the feasibility of short-term future spot prices prediction, which can improve the cost-effectiveness of any cloud processing bringing up big

savings comparing to the on-demand prices. This way, we provide a significantly different application of the presented framework for multi-stream feature extraction and analysis. Instead of referring to multi-dimensional data representation, we performed univariate analysis of many time series independently where the feature extraction part was limited to extracting candlesticks from sliding windows over the spot price bidding data collected from AWS Cloud [8, 145]. The main reason behind the evaluation of ARIMA models on data represented as candlestick is that both techniques are very popular and easily interpretable by experts.

5.4.1 Introduction

Proper allocation of cloud resources is a challenging task, particularly for computationally cumbersome tasks like data processing. There are quite a few examples of cluster size optimizations for Big Data analytics that focus on resource management for sustainable and reliable cloud computing [125]. One of the approaches could rely on initial estimations of data stream characteristics expressed in a vector termed Characteristics of Data (CoD). Clusters of cloud resources could then be created dynamically with the help of, e.g., Self-Organizing Maps [205, 255]. Another approach – presented in [106] – focuses on the optimization of short-running jobs. Authors in [160] propose a query-like environment where developers can query for the required cluster size. The proposed approach requires, however, implementation-specific details. The evaluation of historic executions and metrics is considered as one of the prominent methods that leads to proper optimization, resulting in the timely processing of data [420].

Cloud providers aim to optimize server utilization to avoid idle capacity and significant peaks [206]. This led to the emergence of cloud spot markets on which service providers and customers can trade computation power in near real-time. One of the evident concerns regarding the spot model is that prices fluctuate along with changes in supply and demand. Furthermore, cloud providers may terminate provisioned instances with a minute notice due to outbidding. The ability to forecast future spot prices in a time horizon necessary to complete the data processing tasks would be a game-changer allowing to decrease total costs of operation of data processing pipelines, and to minimize the risk of resource terminations. In practice, typical data processing tasks, and in particular feature engineering tasks, have a degree of temporal flexibility - they need to be fulfilled before a specified deadline. However, it is often possible to defer the computations if it could lead to overall cost reduction due to price fluctuation. With a reliable forecasting model that provides accurate spot price prediction for a given time horizon, and reliable estimation of resources required to perform the task, one could recommend an efficient and cost-effective cluster configuration.

Some of the frameworks for cluster size optimization, to minimize the deployment cost, consider allocating server time to spot cloud resources. For that purpose, a fine-tuned heuristic to automate application deployment, and a Markov model that describes the stochastic evolution of the spot price and its influence on virtual machine reliability are proposed [99]. In [400], the authors describe an integral framework for sharing time on servers between on-demand and spot services. This is one way to

guarantee that on-demand users can be served quickly while spot users can stably use servers for an appropriately long period. This is a critical feature in making both on-demand and spot services accessible. However, guaranteeing timely cloud job execution on a spot instance is a very challenging task, and existing strategies may not fulfill requests in case of outbidding.

Changes in supply and demand are the primary factor that impacts the price of a given service. This behavior is well-known in the stock exchange or commodity markets [54, 110]. Among many available methods for time series regression [337] – which are the most suitable for modeling the problem of price prediction – one of the most popular and broadly used are autoregressive integrated moving average (ARIMA) models [4]. Results obtained in this study confirmed that ARIMA has a strong potential for short-term spot prediction.

The ability to accurately forecast future spot prices is essential to minimize the risk of resource terminations. A number of models have already been applied for that task [206]. For example, in [25], the authors evaluated a model to predict EC2 spot prices based on long/short-term memory recurrent neural networks. The problem of forecasting EC2 spot prices one day and one week ahead was also evaluated with random forest regressors [210]. Because of the similarity between cloud spots and financial markets [110], we decided to assess ARIMA models [71], which are known to be robust and efficient in short-term time series forecasting on stock exchanges or commodity markets [54, 83].

5.4.2 Cloud Spot Market

Spot instances can be regarded as spare compute capacity in cloud data centers. They are offered as one of the three ways cloud providers sell their computing capacity – the other two are on-demand and reserved instances. In terms of the servers, there is no difference between the three. The difference is in the business model. On-demand instances represent the pay-as-you-go model, while reserved instances facilitate long-term renting of computing resources with a discount. However, spot instances allow customers to save up to 90% of costs by using the cloud's unused servers (cf. Figure 5.13). The two most popular cloud providers, Amazon AWS[3] and Windows Azure[4], have such spot instance offerings. Even though both Windows Azure and Amazon AWS offer spot instances, there are years of spot instance price history available for AWS. Therefore, most of the discussion later in this section revolves around AWS types of spot instances.

With spot instances, customers never pay more than the maximum price specified in the bid. However, the evident concern with the spot model is that the cloud provider may terminate these instances with literally last-minute notice. AWS offers various options to configure interruption behavior of Spot instances and Spot fleets (a set of spot instances), including hibernation and automatic restarting. When the AWS Spot service determines to hibernate a Spot Instance, an interruption notice is issued as a CloudWatch event, but the customer does not have time before the Spot Instance is interrupted, and hibernation begins immediately. To prevent

[3] aws.amazon.com/ec2/spot/

[4] azure.microsoft.com/en-us/pricing/spot/

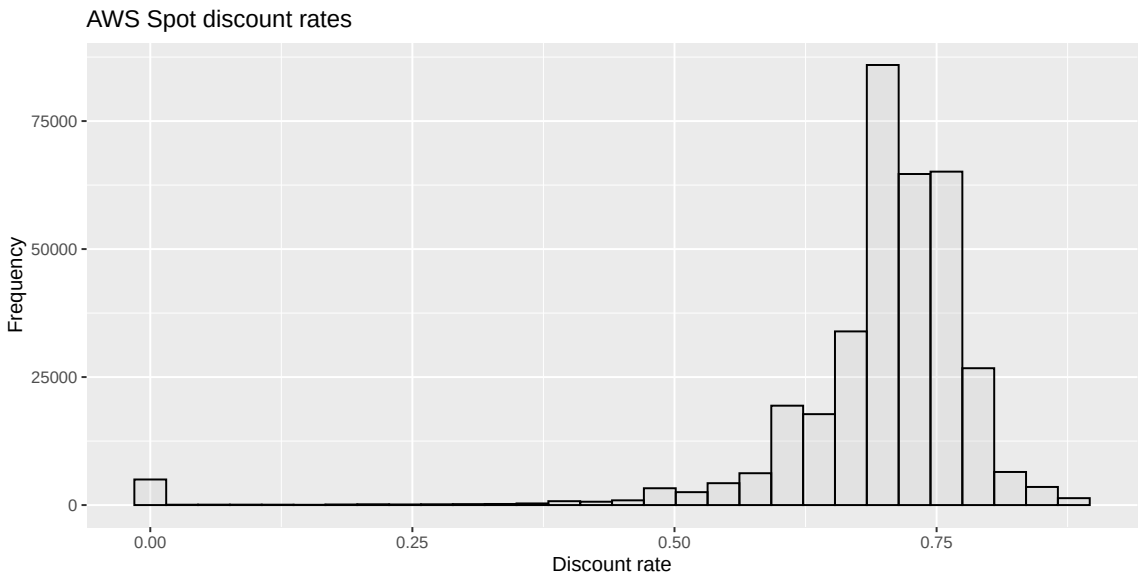


Figure 5.13: Histogram of discount rates for the Linux/UNIX spot machines compared to on-demand pricing. The values are computed based on the data described in Section 5.4.4

interruptions, the best practices suggest using the on-demand price for bidding, storing necessary data regularly at persistent storage (e.g., Amazon S3, Amazon EBS, or DynamoDB), and dividing the work into small tasks while using checkpoints. The more advanced techniques consider future spot price prediction this, in turn, allow to either avoid resource termination by bidding higher, or to store partial results before interruption is triggered by cloud provider.

5.4.3 Univariate Prediction Methods

Naive predictions As in many short-term forecast problems, the last known value is a reasonably good indicator of the next value. Thus, such predictions are commonly used as a baseline. In this section, we refer to it as *Naive* prediction model – that refers to the last known spot price of a particular instance type in the given availability zone. With this approach, at the time when we need to predict future spot price of a particular instance, we simply use the current price and predict that all future prices are going to be equal to it. Obviously, this is a very naive assumption, completely ignoring the dynamic demand for spot instances. As the prediction is about values further away in the future, the expectation is that the quality of such a forecast would significantly decrease. Still, the motivation to include this approach in the evaluation is derived from the exploratory analysis of the dataset showing that the spot prices of some instances were infrequently changing.

Autoregressive integrated moving average ARIMA form a class of time series models that are widely applicable in the field of time series forecasting. ARIMA models are known to be robust and efficient in short-term time series forecasting, with some prominent results in financial and commodity markets, or for anomaly detection in IoT environments [71, 83]. In the ARIMA model, the future value of a

Table 5.10: The most popular spot machines.

region	az	machine	N	bid freq.(h)		bid price ($)	
				Avg	StDev	Avg	StDev
ap-northeast-2	a	r4.8xlarge	513	5.609858	1.850305	0.61898109	0.032847797
ap-south-1	c	m5.4xlarge	511	5.646654	2.098600	0.26793268	0.049972972
ap-south-1	a	m4.10xlarge	510	5.664042	1.919549	0.64866667	0.065327196
us-west-1	a	r4.8xlarge	509	5.679289	1.691382	0.68227269	0.042938466
us-west-1	a	r5.4xlarge	507	5.697051	1.577393	0.56153195	0.082851776
sa-east-1	c	m4.4xlarge	505	5.718381	2.680653	0.33331168	0.022590959
us-west-2	b	c5n.4xlarge	504	5.720918	1.442650	0.40579861	0.031436667
us-west-1	b	m5d.4xlarge	504	5.730931	1.736511	0.37877837	0.071149562
eu-central-1	c	c5d.9xlarge	504	5.732888	1.632358	0.64598294	0.032545920
ap-south-1	b	c5.2xlarge	504	5.740978	1.712241	0.19892956	0.022324986

(a) r5.12xlarge **(b)** r4.16xlarge

Figure 5.14: Candlestick charts for two popular machine types in N. Virginia region, both in AZ: 'b'

variable is a linear combination of past values and errors after removing the trend – by differencing. Given a time series data Y_t where t is an integer index, an ARMA(p,q) model is given by:

$$Y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

where Y_t and ε_t are the actual value and error at time period t, respectively. Whereas, c is a constant, θ_i and φ_i are model parameters to be estimated in the process of model training. ARIMA(p,d,q) model is an extension of ARMA that aims to model non-stationary processes. When the observed time series has a trend, the difference between consecutive observations is computed d times until the observed process becomes stationary.

To provide high-quality spot price forecasts, we trained ARIMA models separately for each AWS instance type in each availability zone. To adjust price prediction to the still changing environment on the AWS spot market, hence minimizing the effect of so-called concept drifts [240], models were iteratively re-trained after each day. The more detailed analysis of spot price prediction is provided further in Section 5.4.6.

5.4.4 Dataset

Contributing to the popularity in industry and research community of the Amazon Web Services (AWS), and the hardware heterogeneity offered in various instance types [259], AWS was used for the experimental evaluation. AWS cloud consists of geographically dispersed regions around the world, each with multiple availability zones (AZ's)[5]. In each of the regions, AWS offers a broad number of cloud services, among which the Elastic Cloud Compute (EC2) is the essential one.

The analyzed spot price data were collected from 11 AWS regions[6]: Tokyo (*ap-northeast-1*), Seoul (*ap-northeast-2*), Mumbai (*ap-south-1*), Singapore (*ap-southeast-1*), Sydney (*ap-southeast-2*), Canada (*ca-central-1*), Frankfurt (*eu-central-1*), Ireland (*eu-west-1*), São Paulo (*sa-east-1*), N. Virginia (*us-east-1*), N. California (*us-west-1*), and Oregon (*us-west-2*) over the period between November 11, 2019 and March 11, 2020. After the preliminary data filtering, we left only those records, which referred to the EMR compatible EC2 machine types – i.e., dedicated for Big Data processing[7] – working with *Linux/UNIX* operating system. For spot instances there is also a constraint that the root volume must be an Elastic Block Store (EBS) volume, not an instance store volume, which eliminated some of the instances from this study. In our study, we were interested in ETL related servers, that is: the memory oriented machine types – m and r series; the computation oriented – c series, and g and p series which are popular for the data analysis and machine learning. The remaining instance families were ignored.

Concerning the savings that could be made with spot instances compared to the corresponding on-demand prices, the prelimynary data exploration results confirmed that the advertised claim of savings up to 90% was indeed true, as shown in Figure 5.13. In Table 5.10, we also present a brief overview of 10 spot price time series for the most popular (i.e., with the most frequent spot price changes) machine types. For more information about the data and the process of data acquisition, we may refer to Appendix A.4.

5.4.5 Data Exploratory Analysis

To verify the feasibility of short term spot price prediction, we decided to limit the scope of analysis further and to focus on the time series with non-trivial price change characteristics. Therefore, we discarded machines with infrequent bids (less than 100 bids in the entire data set), as well as the time series with almost constant prices in the analyzed time range (with the standard deviation of prices $\sigma < 0.01$). The final data set contained 854 time series for 85 different machine types aggregated in non-overlapping candlesticks – a standard tool in financial stock market analysis [8]. Candlestick charts are often used together with various machine learning models, like SVM or DNN [222]. In the performed experiments, each candlestick contained volume of operations as well as open, high, low, and close price during one day. The exemplary candlestick charts for two popular machine types in North Virginia

[5] See AWS global cloud infrastructure at `aws.amazon.com/about-aws/global-infrastructure`
[6] AWS code-names for regions in brackets.
[7] See `docs.aws.amazon.com/emr/latest/ManagementGuide/emr-supported-instance-types`

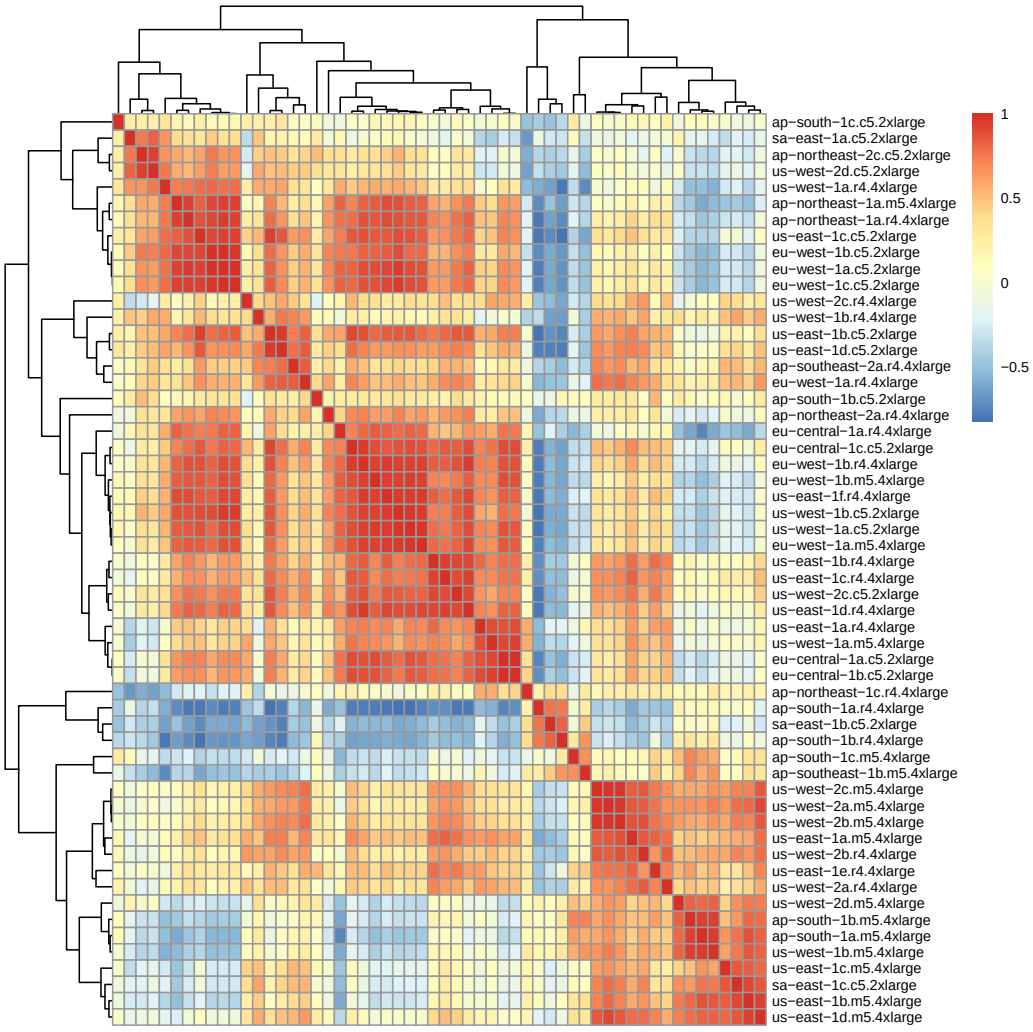
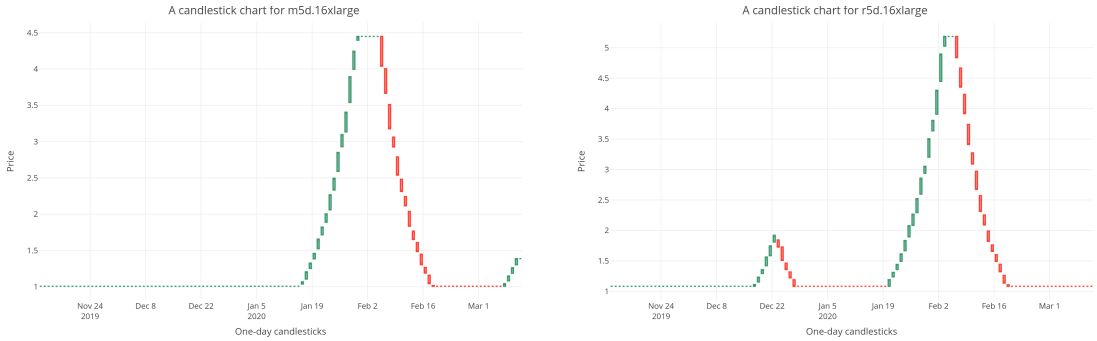


Figure 5.15: Correlation heatmap for the three popular spot machine types: r4.4xlarge, m5.4xlarge, c5.2xlarge.

(*us-east-1*) region are depicted in Figure 5.14. In Figure 5.17, the schematic flow of the entire data collection and machine learning process is presented.

5.4.6 Spot Price Predictions

Having the data aggregated in one-day candlesticks, for a given day (t_x), we aim to predict the highest price during the next day (t_{x+1}). The models are evaluated with two commonly used error metrics, namely, root mean square error (RMSE) and mean absolute percentage error (MAPE). However, to make the prediction and obtained error rates comparable between various machines, the prices were scaled – they were divided by the on-demand price of the same machine type in the corresponding region. This allows providing an estimation of a budget needed for the data processing task. The preliminary analysis showed that even the *Naive* model, which used as a prediction the last day price, achieved a relatively good quality. The highest MAPE of 5.49% was recorded for the *m5d.16xlarge* machine type in *ap-northeast-2* region (See Figure 5.16 (a)). The median and macro average of MAPE over all 854 evaluated



(a) ap-northeast-2(b), machine:m5d.16xl **(b)** us-west-1(b), machine:r5d.16xl

Figure 5.16: Candlestick charts of two machines troublesome for prediction.

time series was 0.66% and 0.87%, respectively. These results mean that in the worst case, we can expect a budget overrun of ca. 5.49% in the event of a rapid price change (as shown in Figure 5.16). However, on average, the error will be much smaller. The results of Naive model performance aggregated over all 854 time series are presented in Table 5.11 in the row signed *Naive*. The median RMSE, in Table 5.11, refers to median value of 854 experiments. Similarly, in the table, we also report macro average, 3^{rd} quantile, and max value for RMSE and MAPE.

In the performed study, we trained ARIMA models for each of 854 time series in data. Similar to the Naive model's case, the evaluation was performed on the last two months in data (60 days). Before each assessment (at time t_x) of next day price, the ARIMA model was re-trained on all available historical data ($t_0 ... t_x$). The aggregated performance of the ARIMA model trained on all available history is presented in Table 5.11 in the row marked as "ARIMA(All)". To further verify the optimal history size for the estimation of model's parameters, that allows more dynamically respond to the shifts in characteristics of AWS spot prices, we repeated the experiments for various length of training data history to fit the ARIMA model (from 10 days to 50 days). In the case of 40 day long history the maximal MAPE error of 3.84% was recorded in *us-west-1* region for *r5d.16xlarge* machine (see Figure 5.16

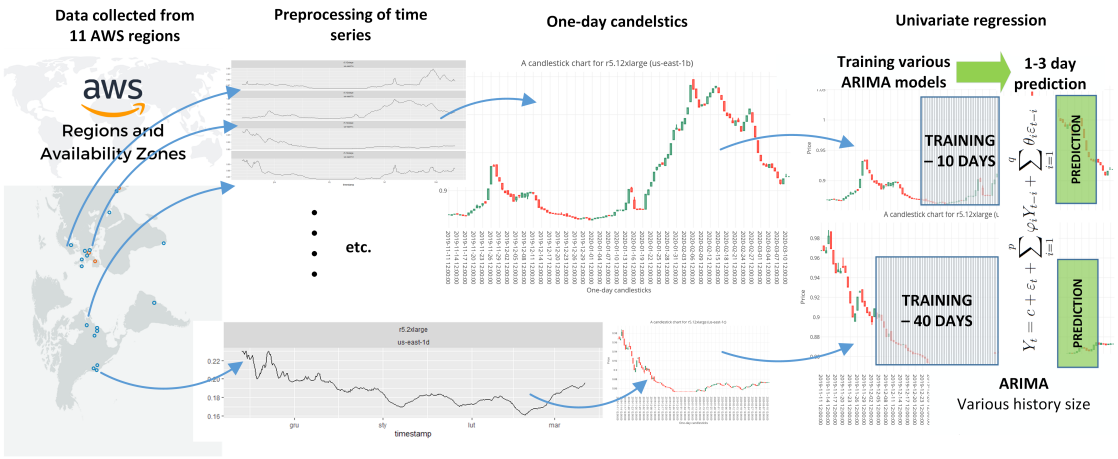


Figure 5.17: A schematic flow of the entire data collection and machine learning process.

Table 5.11: Spot price prediction in one day horizon. ARIMA with various sliding window settings (training history length in brackets) and the naive model with last known value. The minimal error for all classifiers was equal to zero. The 1^{st} quantile was always smaller than 0.001 and 0.15 for R2 and MAPE, respectively. Prediction evaluated on the last two months in data (60 days).

Model	Hist. size	RMSE				MAPE			
		Median	MacAvg	3^{rd}Qu.	Max	Median	MacAvg	3^{rd}Qu.	Max
ARIMA	10	0.00354	0.00624	0.00651	**0.04195**	0.7069	0.9088	1.2520	5.1401
	15	0.00315	**0.0055**	0.0071	0.0617	0.6486	0.8008	1.1552	3.8915
	20	0.00325	0.00643	0.00759	0.05964	0.6620	0.8884	1.2644	5.1581
	25	0.00313	0.00630	0.00750	0.05926	0.6322	0.8597	1.1928	6.4112
	30	0.00305	0.00598	0.00714	0.05909	0.6199	0.8240	1.2005	5.5383
	35	0.00299	0.00606	0.00715	0.08033	0.6101	0.8197	1.1622	4.7605
	40	0.00293	0.00577	0.00688	0.05909	0.5987	0.8020	1.1433	**3.842**
	45	0.00286	0.00577	0.00704	0.07563	0.5954	0.8032	1.1157	4.0203
	50	0.00286	0.00557	0.00692	0.10255	0.5893	**0.789**	1.0645	4.9812
	All	**0.0028**	0.00551	0.00739	0.07884	**0.5866**	0.8051	**1.0335**	6.5449
Naive	1	0.00314	0.00607	**0.0059**	0.04648	0.6640	0.8782	1.2556	5.4873

(b)). This provides us with the worst-case estimation of cost under- or over-run of spot resource allocation. Still, in the (macro) averaged or mean case, we would be far more accurate. In Table 5.11, the aggregated analysis for the Naive model and all ARIMA settings is presented.

For the more in-depth analysis, we decided to select Naive, ARIMA(All), ARIMA(40) models. The first one presents a baseline, the second achieved lowest average errors, whereas the ARIMA(40) minimized the maximal MAPE error, which assures the lowest worst-case budget misestimation. The three main parameters to be estimated in the ARIMA(p, d, q) model are the number of time lags of the auto-regressive model p, degree of differencing d, and the order of the moving average model (q). In our experiments, these parameters were estimated using the Box–Jenkins approach. The analysis of the selected models revealed that for various time series and length of available training data, different p and q parameter values were chosen. In the performed experiments, the series was most often differenced once - trend components for the trained ARIMA models were usually $d = 1$. The AR parameters were typically equal to 1 or 2, whereas MA parameters varied from 0 to 4. The seasonality test was negative in all examined cases. Hence, the trained ARIMA models were of a form ARIMA$(1-2, 1, 0-4)$.

To validate the statistical significance of observed differences between the performance of the selected models, we decided to employ the Wilcoxon signed rank test – due to a very low p-value observed during Shapiro-Wilk normality test on both RMSE and MAPE distributions achieved during the tests. In all the cases, p-value of Shapiro-Wilk normality test was: $p\text{-}value < 1.0e\text{-}15$. In the case of RMSE, the Wilcoxon signed rank test, with the null hypothesis that the errors of the Naive model are not greater than those of ARIMA(40) did not allow to reject this hypothesis ($p\text{-}value = 0.1954$). However, when the ARIMA(All) model was

Table 5.12: A summary of the prediction errors of the selected models for various forecasting horizons (1-3 days).

Model	Pred. day	RMSE				MAPE			
		Median	MacAvg	3^{rd}Qu.	Max	Median	MacAvg	3^{rd}Qu.	Max
ARIMA (All)	1	0.0028	0.0055	0.0074	0.0788	0.5866	0.8051	1.034	6.545
	2	0.0053	0.01	0.0118	0.095	1.15	1.537	2.029	10.73
	3	0.0076	0.0146	0.0152	0.142	1.647	2.28	2.948	16.32
ARIMA (40)	1	0.0029	0.0058	0.0069	0.0591	0.599	0.802	1.143	3.842
	2	0.0056	0.0105	0.0123	0.079	1.187	1.57	2.18	7.96
	3	0.0079	0.015	0.017	0.136	1.72	2.36	3.24	13.98
Naive	1	0.00314	0.0061	0.006	0.0465	0.664	0.878	1.256	5.487
	2	0.00547	0.0112	0.011	0.0887	1.1841	1.6459	2.335	11.01
	3	0.0073	0.0157	0.014	0.126	1.609	2.345	3.317	16.48

compared to ARIMA(40) and Naive, the p-values of both tests were very low, i.e., $3.786e$-08 for Naive and $4.649e-06$ for ARIMA(40), respectively. It allowed us to reject the null hypothesis, hence showing the statistical significance of differences between the models. Slightly different observations were made for the MAPE measure. In this case, the Wilcoxon test revealed that ARIMA(40) model was significantly better than Naive ($p\text{-}value = 3.053e\text{-}07$). However, the ARIMA(All) again performed significantly better than both ARIMA(40) ($p\text{-}value = 0.005515$) and Naive ($p\text{-}value = 3.396e\text{-}11$) models.

In the last part of our study, we attempted to validate the feasibility of spot price prediction in a bit longer horizon of two and three days ahead. We examined the performance of the three selected models from our previous test: Naive, ARIMA(All) and ARIMA(40). The results - presented in Table 5.12 - showed that the observed drop of each model performance is significant, and the maximal MAPE error exceeds 16% for both ARIMA(All) and Naive models. However, we may conclude that prediction is still feasible two and three days ahead, with a median of MAPE errors only slightly exceeding 1.6% for ARIMA(All).

An interesting approach to further investigation would be to use multivariate methods, mainly due to the observed correlations between various time series in multiple regions and availability zones, as shown in Figure 5.15. This figure is a heatmap with a dendrogram added to the left side and to the top where the colour of each cell represents the correlation between the price of a pair of instance types in different availability zones. A dendrogram is a tree-structured graph that visualizes the result of a hierarchical clustering calculation. For the dendrogram on the left side of the heatmap, the individual rows in the clustered data are represented by the right-most nodes (i.e., the leaf nodes). Each node in the dendrogram represents a cluster of all rows from the connected leaves. The left-most node in the dendrogram is therefore a cluster that contains all rows.

Chapter 6

Concluding Remarks and Future Works

This chapter concludes the dissertation and summarizes the presented research. It also indicates some possible research directions for the future development of the interactive feature extraction methods and points out some interesting application areas.

6.1 Summary

In the dissertation, we discuss interactive feature extraction, and we propose several innovative approaches to automating feature creation and selection processes. In the study on the interactiveness of the feature extraction methodologies, we address the problems of deriving relevant and understandable attributes from raw sensor readings and reducing the amount of those attributes to achieve possibly simplest yet accurate models. The proposed algorithms for the construction and selection of features can use various forms of granulation, problem decomposition, and parallelization. Consequently, they respond to the requirements of expressing complex concepts intuitively and efficiently, which are essential for the feasibility of feature selection.

Feature selection is crucial for constructing prediction and classification models, resulting in their higher quality and interpretability. However, the selected features may become temporarily unavailable in a long-term time frame, which can disable a pre-trained model and cause a severe impact on business continuity. The novel methods introduced in the dissertation go beyond the current standards. Accordingly, we formalize the notion of resilient feature selection by introducing r-$\mathbb{C}$-reducts – irreducible subsets of attributes providing a satisfactory level of information about the target variable according to a given criterion function $\mathbb{C}$, even after removing arbitrary r elements. The proposed approach is based on a generalization of (approximate) reducts known from the rough set theory (RST) [367]. The framework proposed in this paper embraces a much wider family of criteria specifying that a given feature subset is good enough to determine target variable values. We are actually able to refer to the whole realm of filter-based feature selection strategies [79], now defining a satisfactory feature set as the one whose evaluation function exceeds a certain threshold even after removing its arbitrary r elements, $r \geq 0$.

We proved that any NP-hard problem of finding a minimal attribute subset that yields a satisfactory level of information according to a given criterion function $\mathbb{C}$ remains NP-hard for an arbitrary resilience level r. As a special case, the task of finding a minimal subset of features providing ε-almost the same level of the aforementioned accuracy measure as the whole set even after removing arbitrary r elements is NP-hard. The dissertation discusses also opportunities of the exhaustive and heuristic search of r-$\mathbb{C}$-reducts. By following a popular idea of dynamic exploration of the lattice of feature subsets, whereby some of its elements turn out to be labeled as satisfying the criteria for providing enough information while others do not, we elaborate on two generic algorithmic strategies, namely: breadth first search (BFS), and depth first search (DFS). For BFS, we adapted the well-known Apriori algorithm [331] for the purpose of r-$\mathbb{C}$-reduct search (Section 3.3). For DFS, we extended standard reduct construction methods [353] to incorporate resilience of generated feature sets (Section 3.5.1). The presented results confirm that the idea is very promising, and resilient feature selection may significantly minimize the risk and impact of data loss on predictive analysis.

With regard to feature engineering, we present a particular take on the challenge of devising a more effective and efficient feature extraction methodology. The main idea behind our approach is to make intelligent use of the information granulation paradigm in the context of aggregating, selecting, and engineering attributes (features/variables/dimensions) that describe the data. The gist is to operate on attribute granules that are formed through the use of various knowledge discovery algorithms, such as, e.g., clustering or interchangeability analysis through heat maps. In many instances, as exemplified by the use cases discussed, granules built over the attribute space may represent semantic relationships that are important for domain experts. The proposed framework facilitates discovering meaningful knowledge from the underlying data, which may be further leveraged in order to obtain a more comprehensible and user-friendly representation that is described in a possibly intuitive way, i.e., using statistics characterizing sliding time windows (Section 4.1.2). In the case of the underground coal mine sensors, derivation of multivariate series of window-based statistics allowed us to deal with noisy and incomplete data sources, better reflected temporal drifts and correlations, and reliably described real situations using higher-level data characteristics.

As a notable aspect and an important research field addressed in the frame of this study, let us point out the framework for linking sliding window-based feature creation, resilient feature selection, and machine learning techniques to build predictive models that are understandable for experts and resistant to partial data loss (Chapter 4). The solution conveys the granular knowledge in the data to the final decision model. At the same time, it is designed to deal with enormous amounts of information that needs to be processed when facing the kinds of tasks typical for Big Data. The proposed methods for feature extraction are easy to maintain and efficient to compute. They are understandable for the system users and domain experts by means of operating with small subsets of intuitively defined features, which is an important aspect from the point of view of interactiveness.

The proposed approaches to interactive feature extraction have been developed based on the experience gained in the course of several research projects in the fields of

processing mutli-sensory streams in various domains, but also textual data analysis [141]. The experimental study in Chapter 5 confirms the quality of the proposed framework, taking into account its subsequent layers of feature creation, selection (Figures 4.5, and 4.6), forecasting models training and ensemble blending (Figure 4.7). The methods have been validated in terms of the quality of the obtained features, throughput, scalability, and resilience of their operation. The discussed methodology has been successfully applied in several real-life problems related to the time series data [136, 141, 179, 216, 356, 420]. Furthermore, we describe a series of international data mining challenges organized to facilitate this study.

The dissertation addresses a number of challenges related, among others, to the comprehensible and concise representation of the analyzed data or the possibility of embedding domain knowledge into the data. The investigated problems have been thoroughly considered both from the theoretical and practical sides. The developed solutions have been meticulously evaluated in terms of various qualitative aspects like the diversity of the solution or its resilience to data deficiencies. The dissertation provides a comprehensive rationale for this research direction, building a solid theoretical foundation for further considerations related to the interactivity of the feature extraction and machine learning process.

6.2 Future Works

The next steps towards practical use of the outlined methodology would be to devise methods and tools that automate this process and, at the same time, maintain an acceptable level of transparency and human readability in a possibly visual way while taking into account various constraints [52, 277]. Further research on the interactive incorporation of domain knowledge into feature extraction is also a desired direction. For example, in one of the possible scenarios, an analyst collaborates with a feature selection algorithm through a specially designed user interface. In an iterative way, the analyst passes feedback on the relevance of attributes proposed by the algorithm. Therefore, allows to limit the scope of the analysis and improve the quality of the obtained features. A complete system capable of flexible, comprehensible, and extensible interaction with a data scientist who analyzes massive data sets and provides their input by interruption to the extraction process would be an invaluable tool [358].

No less important is integration with the existing technologies. In the presented study, we have shown how the MapReduce principles can be employed. There are, however, many more other techniques that were developed over the years with Big Data in mind. For example, it could be helpful to integrate the proposed methods with the existing tools for the management of massive relational data sets (such as Apache Hive or some approximate database engines) [355]. This way, we could embed the "zoom in/out" operations on attributes into a convenient RDBMS environment.

It would be valuable to continue the study on problems related to monitoring natural hazards in underground coal mines. In particular, to continue work on extending and better utilizing information registered while adding new data sources. There are many places where such information could be useful to configure the steps of consistent data ingestion, preprocessing, and learning forecasting models. It is

commonly known that the stage of feature extraction should take into account the semantics of both the overall forecasting task and particular inputs that may help to build its solution. One of the potentially valuable capabilities would be to use it to customize window-based and inter window-based aggregations applied for particular sensors and groups of sensors during the process of feature extraction. In the case of multi-sensor analytics in the domain of underground coal mining, it may mean applying different feature extraction strategies for different types of sensors, as well as constructing higher-level features basing on spatial information represented by the corresponding mining schemes, such as the one shown in Figure 5.2.

As for future research in this area, it is important to perform a deeper analysis of errors made by different models to identify factors influencing wrong predictions for different mining sites. It is important to continue research on the problem of reliability of prediction models in cases when some of the sensory devices which gather the input data are malfunctioning. This is a very common situation in active mines due to a very harsh working environment. In order to construct prediction models which are robust and insensitive to gaps in incoming data, we would like to develop methods for automatic detection of exchangeable features (i.e., attributes whose values come from different sources but express very similar information). We plan to further investigate feature subset selection methods that keep a controllable degree of information redundancy.

In this particular area, the notion of r-$\mathbb{C}$-reduct can be regarded as a feature selection method derived from the theory of rough sets that allows constructing small subsets of features while maintaining the discernibility of objects in a data set, even in a case when we suspect that a part of attributes may not be available in future. It is, however, worth remembering that there are also some cases of criterion functions $\mathbb{C}$, which model data-based information encoded by attribute subsets in a not (strictly) monotonic way, which means that smaller attribute sets can potentially yield a higher level of information. As a special case, we could consider classes of functions with relaxed monotonicity conditions. In particular, it would be valuable to study functions providing weak, quasi, and directional monotonicity [297, 335]. We could also consider various aggregation functions (called also pre-aggregation functions), directionally increasing conjunctors and implications [47], or mixture functions - a type of weighted averages for which the corresponding weights are calculated by means of appropriate continuous functions of their inputs, which need not be monotone increasing [385].

It is particularly important to investigate all the above aspects in both theoretical and empirical study, including the assessments in real-life environments, taking into account simulations based on pessimistic and random attribute removal scenarios that model temporary, partial unavailability of data sources. For the same reason, it is needed to empirically compare the proposed framework with other (both RST-based and not RST-based) approaches to stable and robust feature selection [3, 198].

We do believe that further research results may have a significant impact on the development of feature extraction. The presented future research directions have a solid practical motivation. Deriving meaningful features, which are interpretable for human experts, is important in many domains, as medicine, criminal justice, or industrial processes monitoring [119, 325, 356]. Such methods may increase the

safety of people working in, e.g., underground mines or participating in firefighting rescue operations. Furthermore, through analysis of variables' importance [111] and co-predictive mechanisms between interpretable features [118], they may foster the understanding of the root causes of modeled phenomena.

Bibliography

[1] Mohamed Abdel-Basset, Doaa El-Shahat, Ibrahim M. El-Henawy, Victor Hugo C. de Albuquerque, and Seyedali Mirjalili. A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst. Appl.*, 139, 2020.

[2] Ziawasch Abedjan, Nozha Boujemaa, Stuart Campbell, Patricia Casla, Supriyo Chatterjea, Sergio Consoli, Cristobal Costa-Soria, Paul Czech, Marija Despenic, Chiara Garattini, Dirk Hamelinck, Adrienne Heinrich, Wessel Kraaij, Jacek Kustra, Aizea Lojo, Marga Martin Sanchez, Miguel A. Mayer, Matteo Melideo, Ernestina Menasalvas, Frank Moller Aarestrup, Elvira Narro Artigot, Milan Petković, Diego Reforgiato Recupero, Alejandro Rodriguez Gonzalez, Gisele Roesems Kerremans, Roland Roller, Mario Romao, Stefan Ruping, Felix Sasaki, Wouter Spek, Nenad Stojanovic, Jack Thoms, Andrejs Vasiljevs, Wilfried Verachtert, and Roel Wuyts. *Data Science in Healthcare: Benefits, Challenges and Opportunities*, pages 3–38. Springer International Publishing, Cham, 2019.

[3] Thomas Abeel, Thibault Helleputte, Yves Van de Peer, Pierre Dupont, and Yvan Saeys. Robust Biomarker Identification for Cancer Diagnosis with Ensemble Feature Selection Methods. *Bioinformatics*, 26(3):392–398, 2010.

[4] Ayodele Ariyo Adebiyi, Aderemi Oluyinka Adewumi, and Charles K. Ayo. Comparison of ARIMA and artificial neural networks models for stock price prediction. *J. Applied Mathematics*, 2014:614342:1–614342:7, 2014.

[5] United States Fire Administration. Annual report on firefighter fatalities in the united states. http://apps.usfa.fema.gov/firefighter-fatalities/.

[6] Charu C. Aggarwal, editor. *Managing and Mining Sensor Data*. Springer, 2013.

[7] Ankur Agrawal, Jungwook Choi, Kailash Gopalakrishnan, Suyog Gupta, Ravi Nair, Jinwook Oh, Daniel A. Prener, Sunil Shukla, Vijayalakshmi Srinivasan, and Zehra Sura. Approximate computing: Challenges and opportunities. In *IEEE International Conference on Rebooting Computing, ICRC 2016, San Diego, CA, USA, October 17-19, 2016*, pages 1–8. IEEE Computer Society, 2016.

[8] Elham Ahmadi, Milad Jasemi, Leslie Monplaisir, Mohammad Amin Nabavi, Armin Mahmoodi, and Pegah Amini Jam. New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the support vector machine and heuristic algorithms of imperialist competition and genetic. *Expert Systems with Applications*, 94:21 – 31, 2018.

[9] Farrukh Ahmed, Michele Samorani, Colin Bellinger, and Osmar R. Zaïane. Advantage of Integration in Big Data: Feature Generation in Multi-Relational Databases for Imbalanced Learning. In *Proc. of IEEE BigData 2016*, pages 532–539.

[10] Selim Aksoy and Robert M. Haralick. Feature Normalization and Likelihood-based Similarity Measures for Image Retrieval. *Pattern Recogn. Lett.*, 22(5):563–582, April 2001.

[11] Hamda Al-Ali, Alfredo Cuzzocrea, Ernesto Damiani, Rabeb Mizouni, and Ghalia Tello. A composite machine-learning-based framework for supporting low-level event logs to high-level business process model activities mappings enhanced by flexible BPMN model translation. *Soft Comput.*, 24(10):7557–7578, 2020.

[12] Salem Alelyani, Jiliang Tang, and Huan Liu. Feature selection for clustering: A review. In Charu C. Aggarwal and Chandan K. Reddy, editors, *Data Clustering: Algorithms and Applications*, pages 29–60. CRC Press, 2013.

[13] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *CoRR*, abs/1901.09069, 2019.

[14] Wilker Altidor, Taghi M. Khoshgoftaar, and Amri Napolitano. Measuring Stability of Feature Ranking Techniques: A Noise-based Approach. *International Journal of Business Intelligence and Data Mining*, 7(1-2):80–115, 2012.

[15] Annalisa Appice, Pietro Guccione, Donato Malerba, and Anna Ciampi. Dealing with Temporal and Spatial Correlations to Classify Outliers in Geophysical Data Streams. *Information Sciences*, 285:162–180, 2014.

[16] Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. Big Data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, 79:3–15, 2015.

[17] Piotr Augustyniak, Magdalena Smoleń, Zbigniew Mikrut, and Eliasz Kańtoch. Seamless tracing of human behavior using complementary wearable and house-embedded sensors. *Sensors*, 14(5):7831–7856, 2014.

[18] Shaeela Ayesha, Muhammad Kashif Hanif, and Ramzan Talib. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, 59:44 – 58, 2020.

[19] Mohammad Azad and Mikhail Moshkov. Minimization of decision tree average depth for decision tables with many-valued decisions. *Procedia Computer Science*, 35:368 – 377, 2014. Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.

[20] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable K-Means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.

[21] Magdalena Bałazińska and Stan Zdonik. *Databases Meet the Stream Processing Era*, page 225–234. Association for Computing Machinery and Morgan and Claypool, 2018.

[22] Ankita Bansal, Roopal Jain, and Kanika Modi. *Big Data Streaming with Spark*, pages 23–50. Springer Singapore, Singapore, 2019.

[23] Andrzej Bargiela and Witold Pedrycz. The roots of granular computing. In *2006 IEEE International Conference on Granular Computing*, pages 806–809. IEEE, 2006.

[24] Faraj Bashir and Hua-Liang Wei. Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. *Neurocomputing*, 276:23–30, 2018. Machine Learning and Data Mining Techniques for Medical Complex Data Analysis.

[25] Matt Baughman, Christian Haas, Rich Wolski, Ian Foster, and Kyle Chard. Predicting Amazon Spot Prices with LSTM Networks. In *Proceedings of the 9th Workshop on Scientific Cloud Computing*, ScienceCloud'18, page 7, New York, NY, USA, 2018. Association for Computing Machinery.

[26] Jan G. Bazan. Hierarchical Classifiers for Complex Spatio-temporal Concepts. *Trans. Rough Sets*, 9:474–750, 2008.

[27] Jan G. Bazan, Stanisława Bazan-Socha, Sylwia Buregwa-Czuma, Łukasz Dydo, Wojciech Rząsa, and Andrzej Skowron. A Classifier Based on a Decision Tree with Verifying Cuts. *Fundam. Informaticae*, 143(1-2):1–18, 2016.

[28] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[29] María José Benítez-Caballero, Jesús Medina, Eloísa Ramírez-Poussa, and Dominik Ślęzak. A computational procedure for variable selection preserving different initial conditions. *Int. J. Comput. Math.*, 97(1-2):387–404, 2020.

[30] Sandra Benítez-Peña, Rafael Blanquero, Emilio Carrizosa, and Pepa Ramírez-Cobo. Cost-sensitive Feature Selection for Support Vector Machines. *Comput. Oper. Res.*, 106:169–178, 2019.

[31] Mohamed Bennasar, Yulia Hicks, and Rossitza Setchi. Feature selection using joint mutual information maximisation. *Expert Syst. Appl.*, 42(22):8520–8532, 2015.

[32] Frenay Benoit, Mark van Heeswijk, Yoan Miche, Michel Verleysen, and Amaury Lendasse. Feature selection for nonlinear models with extreme learning machines. *Neurocomputing*, 102:111 – 124, 2013.

[33] A. Berrado and G. C. Runger. Supervised multivariate discretization in mixed data with random forests. In *2009 IEEE/ACS International Conference on Computer Systems and Applications*, pages 211–217, May 2009.

[34] Abdelkader Berrouachedi, Rakia Jaziri, and Gilles Bernard. Deep extremely randomized trees. In Tom Gedeon, Kok Wai Wong, and Minho Lee, editors, *Neural Information Processing*, pages 717–729, Cham, 2019. Springer International Publishing.

[35] Gérard Biau, Benoît Cadre, and Laurent Rouvière. Accelerated gradient boosting. *Mach. Learn.*, 108(6):971–992, 2019.

[36] Albert Bifet, Geoffrey Holmes, Bernhard Pfahringer, Richard Kirkby, and Ricard Gavaldà. New ensemble methods for evolving data streams. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 139–148. ACM, 2009.

[37] Verónica Bolón-Canedo and Amparo Alonso-Betanzos. Ensembles for feature selection: A review and future trends. *Inf. Fusion*, 52:1–12, 2019.

[38] Benjamin M. Bolstad, Rafael A. Irizarry, Magnus Åstrand, and Terence P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[39] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, and Michel Lang. Benchmark for filter methods for feature selection in high-dimensional classification data. *Comput. Stat. Data Anal.*, 143, 2020.

[40] Howard D. Bondell and Brian J. Reich. Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR. *Biometrics*, 64(1):115–123, 2008.

[41] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 5:216–233, September 2015.

[42] Marc Boullé. Modl: A bayes optimal discretization method for continuous attributes. *Mach. Learn.*, 65(1):131–165, October 2006.

[43] Marc Boullé. Prediction of Methane Outbreak in Coal Mines from Historical Sensor Data under Distribution Drift. In Yiyu Yao, Qinghua Hu, Hong Yu, and Jerzy W. Grzymała-Busse, editors, *Proceedings of RSFDGrC 2015*, volume 9437 of *Lecture Notes in Computer Science*, pages 439–451. Springer, 2015.

[44] Marc Boullé. Predicting dangerous seismic events in coal mines under distribution drift. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Proceedings of FedCSIS 2016*, pages 227–230. IEEE, 2016.

[45] Afef Ben Brahim and Mohamed Limam. Robust Ensemble Feature Selection for High Dimensional Data Sets. In *Proceedings of HPCS 2013*, pages 151–157, 2013.

[46] Renato Bruni, Cinzia Daraio, and Davide Aureli. Imputation techniques for the reconstruction of missing interconnected data from higher educational institutions. *Knowledge-Based Systems*, 212:106512, 2021.

[47] Humberto Bustince, Radko Mesiar, Anna Kolesárová, Graçaliz Pereira Dimuro, Javier Fernández, Irene Díaz, and Susana Montes. On some classes of directionally monotone functions. *Fuzzy Sets and Systems*, 386:161–178, 2020. Aggregation Operations.

[48] Wu Cai, Linming Dou, Guangyao Si, Anye Cao, Siyuan Gong, Guifeng Wang, and Shasha Yuan. A new seismic-based strain energy methodology for coal burst forecasting in underground coal mines. *International Journal of Rock Mechanics and Mining Sciences*, 123:104086, 2019.

[49] Alberto Cano and Bartosz Krawczyk. Kappa updated ensemble for drifting data stream mining. *Mach. Learn.*, 109(1):175–218, 2020.

[50] Lijuan Cao, Kok Seng Chua, W. K. Chong, H. P. Lee, and Q. M. Gu. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1):321–336, 2003.

[51] Paris Carbone, Stephan Ewen, Gyula Fóra, Seif Haridi, Stefan Richter, and Kostas Tzoumas. State Management in Apache Flink®: Consistent Stateful Distributed Stream Processing. *Proc. VLDB Endow.*, 10(12):1718–1729, 2017.

[52] Emilio Carrizosa, Vanesa Guerrero, and Dolores Romero Morales. On mathematical optimization for the visualization of frequencies and adjacencies as rectangular maps. *Eur. J. Oper. Res.*, 265(1):290–302, 2018.

[53] Rasim Çekik and Alper Kursat Uysal. A novel filter feature selection method using rough set for short text data. *Expert Syst. Appl.*, 160:113691, 2020.

[54] Zhongpei Cen and Jun Wang. Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer. *Energy*, 169:160 – 171, 2019.

[55] Mariela Cerrada, René-Vinicio Sánchez, Diego Cabrera, Grover Zurita, and Chuan Li. Multi-stage feature selection by using genetic algorithms for fault diagnosis in gearboxes based on vibration signal. *Sensors*, 15(9):23903–23926, 2015.

[56] Debasrita Chakraborty, Vaasudev Narayanan, and Ashish Ghosh. Integration of deep feature extraction and ensemble learning for outlier detection. *Pattern Recognit.*, 89:161–171, 2019.

[57] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *CoRR*, abs/1901.03407, 2019.

[58] Raghavendra Chalapathy, Nguyen Lu Dang Khoa, and Sanjay Chawla. Robust deep learning methods for anomaly detection. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3507–3508, New York, NY, USA, 2020. Association for Computing Machinery.

[59] Girish Chandrashekar and Ferat Sahin. A Survey on Feature Selection Methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[60] Agnieszka Chądzyńska-Krasowska, Paweł Betliński, and Dominik Ślęzak. Scalable Machine Learning with Granulated Data Summaries: A Case of Feature Selection. In *Proc. of ISMIS 2017*, pages 519–529.

[61] Jinxing Che, Youlong Yang, Li Li, Xuying Bai, Shenghu Zhang, and Chengzhi Deng. Maximum relevance minimum common redundancy feature selection for nonlinear data. *Information Sciences*, 409-410:68 – 86, 2017.

[62] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[63] Shu-Heng Chen and Ye-Rong Du. *Granularity in Economic Decision Making: An Interdisciplinary Review*, pages 47–71. Springer International Publishing, 2015.

[64] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[65] Weiwei Cheng, Krzysztof Dembczyński, and Eyke Hüllermeier. Graded multilabel classification: The ordinal case. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 223–230. Omnipress, 2010.

[66] Bogdan S. Chlebus and Sinh Hoa Nguyen. On Finding Optimal Discretizations for Two Attributes. In Lech Polkowski and Andrzej Skowron, editors, *Rough Sets and Current Trends in Computing*, pages 537–544, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[67] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014.

[68] Cheng-Tao Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y. Ng, and Kunle Olukotun. Map-reduce for Machine Learning on Multicore. In *Proc. of NIPS 2006*, pages 281–288.

[69] Davide Ciucci and Yiyu Yao. Synergy of granular computing, shadowed sets, and three-way decisions. *Inf. Sci.*, 508:422–425, 2020.

[70] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

[71] Andrew A. Cook, Göksel Misirli, and Zhong Fan. Anomaly Detection for IoT Time-Series Data: A Survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2020.

[72] Chris Cornelis, Richard Jensen, Germán Hurtado Martín, and Dominik Ślęzak. Attribute Selection with Fuzzy Decision Reducts. *Information Sciences*, 180(2):209–224, 2010.

[73] Laure Crochepierre, Lydia Boudjeloud-Assala, and Vincent Barbesant. Interpretable dimensionally-consistent feature extraction from electrical network sensors, 2020.

[74] Lakshmipadmaja D and B. Vishnuvardhan. Classification performance improvement using random subset feature selection algorithm for data mining. *Big Data Research*, 12:1 – 12, 2018.

[75] Zaineb Chelly Dagdia, Christine Zarges, Gaël Beck, and Mustapha Lebbah. A scalable and effective rough set theory-based approach for big data pre-processing. *Knowl. Inf. Syst.*, 62(8):3321–3386, 2020.

[76] Jianhua Dai and Qing Xu. Approximations and Uncertainty Measures in Incomplete Information Systems. *Information Sciences*, 198:62–80, 2012.

[77] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ECCV'06, pages 428–441, Berlin, Heidelberg, 2006. Springer-Verlag.

[78] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google News Personalization: Scalable Online Collaborative Filtering. In *Proc. of WWW 2007*, pages 271–280.

[79] Sanmay Das. Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection. In *Proceedings of ICML 2001*, pages 74–81, 2001.

[80] Manoranjan Dash and Huan Liu. Consistency-based Search in Feature Selection. *Artificial Intelligence*, 151(1-2):155–176, 2003.

[81] Pradipta K. Dash, Maya Nayak, Manas Ranjan Senapati, and Ian W. C. Lee. Mining for Similarities in Time Series Data Using Wavelet-based Feature Vectors and Neural Networks. *Eng. Appl. Artif. Intell.*, 20(2):185–201, March 2007.

[82] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining Stream Statistics over Sliding Windows. *SIAM Journal on Computing*, 31(6):1794–1813, 2002.

[83] Sergio Adriani David, J. A. Tenreiro Machado, Lucas R. Trevisan, Cláudio M. C. Inácio, and António M. Lopes. Dynamics of Commodities Prices: Integer and Fractional Models. *Fundam. Inform.*, 151(1-4):389–408, 2017.

[84] Umeshwar Dayal, Malu Castellanos, Alkis Simitsis, and Kevin Wilkinson. Data integration flows for business intelligence. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 1–11, New York, NY, USA, 2009. ACM.

[85] Roberto Souto Maior de Barros and Silas Garrido T. de Carvalho Santos. An overview and comprehensive comparison of ensembles for concept drift. *Information Fusion*, 52:213 – 244, 2019.

[86] Marcílio Carlos Pereira de Souto, Ivan G. Costa, Daniel S. A. de Araujo, Teresa Bernarda Ludermir, and Alexander Schliep. Clustering Cancer Gene Expression Data: A Comparative Study. *BMC Bioinformatics*, 9, 2008.

[87] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[88] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[89] Thomas G. Dietterich. Ensemble Methods in Machine Learning. In *Proceedings of MCS 2000*, pages 1–15, 2000.

[90] Chris H. Q. Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.*, 3(2):185–206, 2005.

[91] Ciprian Dobre and Fatos Xhafa. Parallel Programming Paradigms and Frameworks in Big Data Era. *International Journal of Parallel Programming*, 42(5):710–738, Oct 2014.

[92] Patrick Doherty and Andrzej Szalas. Rough set reasoning using answer set programs. *International Journal of Approximate Reasoning*, 130:126–149, 2021.

[93] Pedro Domingos. A Few Useful Things to Know About Machine Learning. *Communications of the ACM*, 55(10):78–87, 2012.

[94] Hongbin Dong, Tao Li, Rui Ding, and Jing Sun. A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Appl. Soft Comput.*, 65:33–46, 2018.

[95] Denis Moreira dos Reis, Peter A. Flach, Stan Matwin, and Gustavo E. A. P. A. Batista. Fast Unsupervised Online Drift Detection Using Incremental Kolmogorov-Smirnov Test. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1545–1554. ACM, 2016.

[96] Linming Dou, Wu Cai, Anye Cao, and Wenhao Guo. Comprehensive early warning of rock burst utilizing microseismic multi-parameter indices. *International Journal of Mining Science and Technology*, 28(5):767 – 774, 2018. SI: Dynamic failure in rock masses.

[97] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, ICML'95, pages 194–202, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[98] Michal Dramiński, Alvaro Rada-Iglesias, Stefan Enroth, Claes Wadelius, Jacek Koronacki, and Henryk Jan Komorowski. Monte Carlo Feature Selection for Supervised Classification. *Bioinformatics*, 24(1):110–117, 2008.

[99] Daniel J. Dubois and Giuliano Casale. OptiSpot: minimizing application deployment cost using spot cloud resources. *Cluster Computing*, 19(2):893–909, 2016.

[100] Didier Dubois and Henri Prade. Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems*, 17, 06 1990.

[101] P. Duda, L. Rutkowski, M. Jaworski, and D. Rutkowska. On the Parzen Kernel-Based Probability Density Function Learning Procedures Over Time-Varying Streaming Data With Applications to Pattern Classification. *IEEE Transactions on Cybernetics*, 50(4):1683–1696, 2020.

[102] Nicolas Duforet-Frebourg, Keurcien Luu, Guillaume Laval, Eric Bazin, and Michael G.B. Blum. Detecting Genomic Signatures of Natural Selection with Principal Component Analysis: Application to the 1000 Genomes Data. *Molecular Biology and Evolution*, 33(4):1082–1093, 12 2015.

[103] Soma Dutta, Andrzej Jankowski, Grzegorz Rozenberg, and Andrzej Skowron. Linking reaction systems with rough sets. *Fundam. Informaticae*, 165(3-4):283–302, 2019.

[104] Carlos Eiras-Franco, Verónica Bolón-Canedo, Sabela Ramos, Jorge González-Domínguez, Amparo Alonso-Betanzos, and Juan Touriño. Multithreaded and Spark Parallelization of Feature Selection Filters. *Journal of Computational Science*, 17:609–619, 2016.

[105] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, and Geoffrey Fox. Twister: A Runtime for Iterative MapReduce. In *Proc. of HPDC 2010*, pages 810–818.

[106] Khaled Elmeleegy. Piranha: Optimizing short jobs in hadoop. *Proceedings of the VLDB Endowment*, 6(11):985–996, 2013.

[107] Jianqing Fan and Jinchi Lv. A Selective Overview of Variable Selection in High Dimensional Feature Space. *Statistica Sinica*, 20(1):101–148, 2010.

[108] Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, pages 1022–1029, 1993.

[109] Junjun Feng, Enyuan Wang, Houcheng Ding, Qisong Huang, and Xia Chen. Deterministic seismic hazard assessment of coal fractures in underground coal mine: A case study. *Soil Dynamics and Earthquake Engineering*, 129:105921, 2020.

[110] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654 – 669, 2018.

[111] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[112] Dorian Florescu and Matthew England. Algorithmically generating new algebraic features of polynomial systems for machine learning. *CoRR*, abs/1906.01455, 2019.

[113] Cristiano Hora Fontes and Otacílio Pereira. Pattern recognition in multivariate time series - a case study applied to fault detection in a gas turbine. *Engineering Applications of Artificial Intelligence*, 49:10–18, 2016.

[114] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[115] Tak-Chung Fu. A Review on Time Series Data Mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[116] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Mach. Learn.*, 73(2):133–153, November 2008.

[117] Wanfu Gao, Liang Hu, and Ping Zhang. Class-specific mutual information variation for feature selection. *Pattern Recognition*, 79:328 – 339, 2018.

[118] Mateusz Garbulowski, Klev Diamanti, Karolina Smolińska, Nicholas Baltzer, Patricia Stoll, Susanne Bornelöv, Aleksander Øhrn, Lars Feuk, and Jan Komorowski. R.ROSETTA: an interpretable machine learning framework. *BMC Bioinform.*, 22(1):110, 2021.

[119] Mateusz Garbulowski, Karolina Smolińska, Klev Diamanti, Gang Pan, Khurram Maqbool, Lars Feuk, and Jan Komorowski. Interpretable machine learning reveals dissimilarities between subtypes of autism spectrum disorder. *Frontiers in Genetics*, 12:73, 2021.

[120] Salvador García, Julián Luengo, José Antonio Sáez, Victoria López, and Francisco Herrera. A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning. *IEEE Trans. Knowl. Data Eng.*, 25(4):734–750, 2013.

[121] Miguel García-Torres, Francisco Gómez-Vela, Belén Melián-Batista, and J. Marcos Moreno-Vega. High-Dimensional Feature Selection via Feature Grouping. *Information Sciences*, 326(C):102–118, 2016.

[122] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[123] Manosij Ghosh, Ritam Guha, Ram Sarkar, and Ajith Abraham. A wrapper-filter feature selection technique based on ant colony optimization. *Neural Comput. Appl.*, 32(12):7839–7857, 2020.

[124] Sławomir J. Gibowicz and Stanisław Lasocki. Seismicity induced by mining: 10 years later. In *Advances in Geophysics*, pages 81–164. 2001.

[125] Sukhpal Singh Gill, Peter Garraghan, Vlado Stankovski, Giuliano Casale, Ruppa K. Thulasiram, Soumya K. Ghosh, Kotagiri Ramamohanarao, and Rajkumar Buyya. Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge. *J. Syst. Softw.*, 155:104–129, 2019.

[126] Alessandro Giuliani. The application of principal component analysis to drug discovery and biomedical data. *Drug Discovery Today*, 22(7):1069 – 1076, 2017.

[127] Wilson Wen Bin Goh and Limsoon Wong. Evaluating feature-selection stability in next-generation proteomics. *J. Bioinform. Comput. Biol.*, 14(5):1–23, 2016.

[128] Igor Goldenberg and Geoffrey I. Webb. Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowl. Inf. Syst.*, 60(2):591–615, 2019.

[129] Jorge González-Domínguez, Roberto R. Expósito, and Verónica Bolón-Canedo. CUDA-JMI: acceleration of feature selection on heterogeneous systems. *Future Gener. Comput. Syst.*, 102:426–436, 2020.

[130] Priya Govindan, Ruobing Chen, Katya Scheinberg, and Soundararajan Srinivasan. A Scalable Solution for Group Feature Selection. In *Proc. of IEEE BigData 2015*, pages 2846–2848.

[131] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. *CoRR*, abs/1802.06893, 2018.

[132] Dominik Grochala, Marcin Kajor, Dariusz Kucharski, Marek Iwaniec, and Eliasz Kańtoch. A novel approach in auscultation technology - new sensors and algorithms. In Adam Bujnowski, Mariusz Kaczmarek, and Jacek Ruminski, editors, *11th International Conference on Human System Interaction, HSI 2018, Gdansk, Poland, July 4-6, 2018*, pages 240–244. IEEE, 2018.

[133] Larry J. Grorud and Dennis Smith. The National Fire Fighter Near-Miss Reporting. Annual Report 2008. *An exclusive supplement to Fire & Rescue magazine*, pages 1–24, 2008.

[134] Alicja Grużdź, Aleksandra Ihnatowicz, and Dominik Ślęzak. Interactive Gene Clustering – A Case Study of Breast Cancer Microarray Data. *Information Systems Frontiers*, 8(1):21–27, 2006.

[135] Tomasz Grychowski. Hazard Assessment Based on Fuzzy Logic. *Archives of Mining Sciences*, 53(4):595–602, 2008.

[136] Marek Grzegorowski. Scaling of Complex Calculations over Big Data-Sets. In *AMT*, volume 8610 of *Lecture Notes in Computer Science*, pages 73–84. Springer, 2014.

[137] Marek Grzegorowski. Governance of the Redundancy in the Feature Selection Based on Rough Sets' Reducts. In Victor Flores, Fernando Gomide, Andrzej Janusz, Claudio Meneses, Duoqian Miao, Georg Peters, Dominik Ślęzak, Guoyin Wang, Richard Weber, and Yiyu Yao, editors, *Rough Sets - International Joint Conference, IJCRS 2016, Santiago de Chile, Chile, October 7-11, 2016, Proceedings*, volume 9920 of *Lecture Notes in Computer Science*, pages 548–557, 2016.

[138] Marek Grzegorowski. Massively Parallel Feature Extraction Framework Application in Predicting Dangerous Seismic Events. In *Proceedings of FedCSIS 2016*, pages 225–229, 2016.

[139] Marek Grzegorowski, Andrzej Janusz, Dominik Ślęzak, and Marcin S. Szczuka. On the Role of Feature Space Granulation in Feature Selection Processes. In Jian-Yun Nie, Zoran Obradovic, Toyotaro Suzumura, Rumi Ghosh, Raghunath Nambiar, Chonggang Wang, Hui Zang, Ricardo Baeza-Yates, Xiaohua Hu, Jeremy Kepner, Alfredo Cuzzocrea, Jian Tang, and Masashi Toyoda, editors, *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 1806–1815. IEEE Computer Society, 2017.

[140] Marek Grzegorowski, Mateusz Kalisch, Michał Kozielski, and Łukasz Wróbel. Hurtowania danych i procesy ETL. In Piotr Przystałka and Marek Sikora, editors, *Zintegrowany, szkieletowy system wspmagania decyzji dla systemów monitorowania procesów, urządzeń i zagrożeń*, chapter 3, pages 31–40. Monograficzna Seria Wydawnicza Instyututu Technik Innowacyjnych EMAG, 2017.

[141] Marek Grzegorowski, Przemysław Wiktor Pardel, Sebastian Stawicki, and Krzysztof Stencel. SONCA: Scalable Semantic Processing of Rapidly Growing Document Stores. In Mykola Pechenizkiy and Marek Wojciechowski, editors, *New Trends in Databases and Information Systems, Workshop Proceedings of the 16th East European Conference, ADBIS 2012, Poznań, Poland, September 17-21, 2012*, volume 185 of *Advances in Intelligent Systems and Computing*, pages 89–98. Springer, 2012.

[142] Marek Grzegorowski and Dominik Ślęzak. On resilient feature selection: Computational foundations of r-C-reducts. *Inf. Sci.*, 499:25–44, 2019.

[143] Marek Grzegorowski and Sebastian Stawicki. Window-Based Feature Engineering for Prediction of Methane Threats in Coal Mines. In Yiyu Yao, Qinghua Hu, Hong Yu, and Jerzy W. Grzymała - Busse, editors, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing - 15th International Conference, RSFDGrC 2015, Tianjin, China, November 20-23, 2015, Proceedings*, volume 9437 of *Lecture Notes in Computer Science*, pages 452–463. Springer, 2015.

[144] Marek Grzegorowski and Sebastian Stawicki. Window-based feature extraction framework for multi-sensor data: A posture recognition case study. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015*, pages 397–405. IEEE, 2015.

[145] Marek Grzegorowski, Eftim Zdravevski, Andrzej Janusz, Petre Lameski, Cas Apanowicz, and Dominik Ślęzak. Cost Optimization for Big Data Workloads Based on Dynamic Scheduling and Cluster-Size Tuning. *Big Data Research*, 25:100203, 2021.

[146] Bin Gu, Guodong Liu, and Heng Huang. Groups-Keeping Solution Path Algorithm for Sparse Regression with Automatic Feature Grouping. In *Proc. of KDD 2017*, pages 185–193.

[147] Yiming Guo, Yifan Zhou, and Zhisheng Zhang. Fault Diagnosis of Multi-channel Data by the CNN with the Multilinear Principal Component Analysis. *Measurement*, page 108513, 2020.

[148] Isabelle Guyon and André Elisseeff. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[149] Isabelle Guyon, Masoud Nikravesh, Steve R. Gunn, and Lotfi A. Zadeh, editors. *Feature Extraction - Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*. Springer, 2006.

[150] Başak Esin Köktürk Güzel and Bilge Karaçalı. Fisher's linear discriminant analysis based prediction using transient features of seismic events in coal mines. In M. Ganzha, L. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, volume 8 of *Annals of Computer Science and Information Systems*, pages 231–234. IEEE, 2016.

[151] Y h. Taguchi. *Unsupervised Feature Extraction Applied to Bioinformatics*. Springer, 2020.

[152] S. Ha and S. Choi. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 381–388, 2016.

[153] Mark Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.

[154] Ahmed Hamed, Ahmed Sobhy, and Hamed Nassar. Distributed approach for computing rough set approximations of big incomplete information systems. *Information Sciences*, 547:427–449, 2021.

[155] Emrah Hancer. Differential evolution for feature selection: a fuzzy wrapper-filter approach. *Soft Comput.*, 23(13):5233–5248, 2019.

[156] Emrah Hancer, Bing Xue, and Mengjie Zhang. Differential evolution for filter feature selection based on information theory and feature ranking. *Knowl. Based Syst.*, 140:103–119, 2018.

[157] Reihaneh H. Hariri, Erik M. Fredericks, and Kate M. Bowers. Uncertainty in big data analytics: survey, opportunities, and challenges. *J. Big Data*, 6:44, 2019.

[158] Yan-Lin He, Ye Tian, Yuan Xu, and Qun-Xiong Zhu. Novel soft sensor development using echo state network integrated with singular value decomposition: Application to complex chemical processes. *Chemometrics and Intelligent Laboratory Systems*, 200, 2020.

[159] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi M. Mafarja, and Huiling Chen. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.*, 97:849–872, 2019.

[160] Herodotos Herodotou, Fei Dong, and Shivnath Babu. No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, page 18. ACM, 2011.

[161] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[162] Kaoru Hirota. Concepts of probabilistic sets. *Fuzzy Sets and Systems*, 5(1):31–46, 1981.

[163] Piotr Hońko. Attribute Reduction: A Horizontal Data Decomposition Approach. *Soft Computing*, 20(3):951–966, 2016.

[164] Babak Hosseini and Barbara Hammer. Interpretable discriminative dimensionality reduction and feature selection on the manifold. In Ulf Brefeld, Élisa Fromont, Andreas Hotho, Arno J. Knobbe, Marloes H. Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I*, volume 11906 of *Lecture Notes in Computer Science*, pages 310–326. Springer, 2019.

[165] Li Hu and Zhiguo Zhang, editors. *EEG Signal Processing and Feature Extraction*. Springer, Singapore, 2019.

[166] Xiaohua Hu. Ensembles of Classifiers Based on Rough Sets Theory and Set-oriented Database Operations. In *Proceedings of IEEE GrC 2006*, pages 67–73, 2006.

[167] Qiang-Sheng Hua, Dongxiao Yu, Francis C. M. Lau, and Yuexuan Wang. Exact Algorithms for Set Multicover and Multiset Multicover Problems. In *Proceedings of ISAAC 2009*, pages 34–44, 2009.

[168] Qingxiang Huang and Jian Cao. Research on coal pillar malposition distance based on coupling control of three-field in shallow buried closely spaced multi-seam mining, china. *Energies*, 12(3), Jan 2019.

[169] Xiaojuan Huang, Li Zhang, Bangjun Wang, Fanzhang Li, and Zhao Zhang. Feature clustering based support vector machine recursive feature elimination for gene selection. *Appl. Intell.*, 48(3):594–607, 2018.

[170] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[171] Rafael A. Irizarry, Bridget Hobbs, Francois Collin, Yasmin D. Beazer-Barclay, Kristen J. Antonellis, Uwe Scherf, and Terence P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[172] Muhammed Tawfiqul Islam, Satish Narayana Srirama, Shanika Karunasekera, and Rajkumar Buyya. Cost-efficient dynamic scheduling of big data applications in apache spark on cloud. *J. Syst. Softw.*, 162, 2020.

[173] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[174] Indu Jain, Vinod Kumar Jain, and Renu Jain. Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Appl. Soft Comput.*, 62:203–215, 2018.

[175] Andrzej Jankowski, Andrzej Skowron, and Roman W. Swiniarski. Interactive complex granules. *Fundam. Inform.*, 133(2-3):181–196, 2014.

[176] Andrzej Janusz. *Algorithms for Similarity Relation Learning from High Dimensional Data*. PhD thesis, University of Warsaw, 2014.

[177] Andrzej Janusz, Łukasz Grad, and Marek Grzegorowski. Clash Royale Challenge: How to Select Training Decks for Win-rate Prediction. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019, Leipzig, Germany, September 1-4, 2019*, volume 18 of *Annals of Computer Science and Information Systems*, pages 3–6, 2019.

[178] Andrzej Janusz, Marek Grzegorowski, Michał Kozielski, Zdzisław Krzystanek, Marek Sikora, Dominik Ślęzak, and Łukasz Wróbel. Przykłady zastosowania systemu DISESOR w analizie i predykcji zagrożeń. In Piotr Przystałka and Marek Sikora, editors, *Zintegrowany, szkieletowy system wspmagania decyzji dla systemów monitorowania procesów, urządzeń i zagrożeń*, chapter 11, pages 31–40. Monograficzna Seria Wydawnicza Instyututu Technik Innowacyjnych EMAG, 2017.

[179] Andrzej Janusz, Marek Grzegorowski, Marcin Michalak, Łukasz Wróbel, Marek Sikora, and Dominik Ślęzak. Predicting Seismic Events in Coal Mines Based on Underground Sensor Measurements. *Engineering Applications of Artificial Intelligence*, 64:83–94, 2017.

[180] Andrzej Janusz, Adam Krasuski, Sebastian Stawicki, Mariusz Rosiak, Dominik Ślęzak, and Hung Son Nguyen. Key Risk Factors for Polish State Fire Service: a Data Mining Competition at Knowledge Pit. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, September 7-10, 2014*, volume 2 of *Annals of Computer Science and Information Systems*, pages 345–354, 2014.

[181] Andrzej Janusz, Marek Sikora, Łukasz Wróbel, Sebastian Stawicki, Marek Grzegorowski, Piotr Wojtas, and Dominik Ślęzak. Mining Data from Coal Mines: IJCRS'15 Data Challenge. In Yiyu Yao, Qinghua Hu, Hong Yu, and Jerzy W. Grzymała-Busse, editors, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing - 15th International Conference, RSFDGrC 2015, Tianjin, China, November 20-23, 2015, Proceedings*, volume 9437 of *Lecture Notes in Computer Science*, pages 429–438. Springer, 2015.

[182] Andrzej Janusz and Dominik Ślęzak. Rough Set Methods for Attribute Clustering and Selection. *Applied Artificial Intelligence*, 28(3):220–242, 2014.

[183] Andrzej Janusz and Dominik Ślęzak. Computation of Approximate Reducts with Dynamically Adjusted Approximation Threshold. In *Proceedings of ISMIS 2015*, volume 9384 of *Lecture Notes in Computer Science*, pages 19–28. Springer, 2015.

[184] Andrzej Janusz, Dominik Ślęzak, Marek Sikora, and Łukasz Wróbel. Predicting Dangerous Seismic Events: AAIA'16 Data Mining Challenge. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016*, volume 8 of *Annals of Computer Science and Information Systems*, pages 205–211. IEEE, 2016.

[185] Andrzej Janusz and Marcin S. Szczuka. Assessment of Data Granulations in Context of Feature Extraction Problem. In *Proc. of IEEE GrC 2014*, pages 116–120.

[186] Andrzej Janusz, Tomasz Tajmajer, and Maciej Świechowski. Helping AI to Play Hearthstone: AAIA'17 Data Mining Challenge. In *Proc. of FedCSIS 2017*, pages 121–125.

[187] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multim. Tools Appl.*, 78(11):15169–15211, 2019.

[188] Xiuyi Jia, Lin Shang, Bing Zhou, and Yiyu Yao. Generalized attribute reduct in rough set theory. *Knowledge-Based Systems*, 91:204–218, 2016. Three-way Decisions and Granular Computing.

[189] Fernando Jiménez, José T. Palma, Gracia Sánchez, David Marín, Francisco Palacios Ortega, and M. D. Lucía López. Feature selection based multivariate time series forecasting: An application to antibiotic resistance outbreaks prediction. *Artificial Intelligence in Medicine*, 104, 2020.

[190] Rong Jin and Luo Si. A Study of Methods for Normalizing User Ratings in Collaborative Filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 568–569, New York, NY, USA, 2004. ACM.

[191] Rong Jin, Luo Si, ChengXiang Zhai, and Jamie Callan. Collaborative Filtering with Decoupled Models for Preferences and Ratings. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 309–316, New York, NY, USA, 2003. ACM.

[192] Yunge Jing, Tianrui Li, Hamido Fujita, Baoli Wang, and Ni Cheng. An incremental attribute reduction method for dynamic data mining. *Inf. Sci.*, 465:202–218, 2018.

[193] Yunge Jing, Tianrui Li, Chuan Luo, Shi-Jinn Horng, Guoyin Wang, and Zeng Yu. An Incremental Approach for Attribute Reduction Based on Knowledge Granularity. *Knowledge-Based Systems*, 104(C):24–38, 2016.

[194] Alan Jovic, Karla Brkic, and Nikola Bogunovic. A Review of Feature Selection Methods with Applications. In *Proceedings of MIPRO 2015*, pages 1200–1205, 2015.

[195] Józef Kabiesz. Effect of the form of data on the quality of mine tremors hazard forecasting using neural networks. *Geotechnical & Geological Engineering*, 24(5):1131–1147, 2006.

[196] Józef Kabiesz. The justification and objective to modify methods of forecasting the potential and assess the actual state of rockburst hazard. In *Methods for assessment of rockburst hazard in coal mines' excavations*, volume 44, pages 44–48. 2010. (in Polish).

[197] Józef Kabiesz, Beata Sikora, Marek Sikora, and Łukasz Wróbel. Application of rule-based models for seismic hazard prediction in coal mines. *Acta Montanistica Slovaca*, 18(3):262–277, 2013.

[198] Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of Feature Selection Algorithms: A Study on High-dimensional Spaces. *Knowledge and Information Systems*, 12(1):95–116, 2007.

[199] Myeongsu Kang and Jing Tian. *Machine Learning: Data Pre-processing*, pages 111–130. 2019.

[200] Eliasz Kańtoch, Piotr Augustyniak, M. Markiewicz, and D. Prusak. Monitoring activities of daily living based on wearable wireless body sensor network. In *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014, Chicago, IL, USA, August 26-30, 2014*, pages 586–589. IEEE, 2014.

[201] Eliasz Kańtoch, Dominik Grochala, Marcin Kajor, and Dariusz Kucharski. The prototype of wearable sensors system for supervision of patient rehabilitation using artificial intelligence methods. In Marek Gzik, Ewaryst Tkacz, Zbigniew Paszenda, and Ewa Piętka, editors, *Innovations in Biomedical Engineering*, pages 205–214, Cham, 2018. Springer International Publishing.

[202] Murat Karabatak and M. Cevdet Ince. A New Feature Selection Method Based on Association Rules for Diagnosis of Erythemato-Squamous Diseases. *Expert Systems with Applications*, 36(10):12500–12505, 2009.

[203] Anusha Kasinikota, P. Balamurugan, and Shirish Shevade. Modeling Label Interactions in Multi-label Classification: A Multi-structure SVM Perspective. In Dinh Phung, Vincent S. Tseng, Geoffrey I. Webb, Bao Ho, Mohadeseh Ganji, and Lida Rashidi, editors, *Advances in Knowledge Discovery and Data Mining*, pages 43–55, Cham, 2018. Springer International Publishing.

[204] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *TKDD*, 6(4):15, 2012.

[205] Navroop Kaur and Sandeep K. Sood. Efficient Resource Management System Based on 4Vs of Big Data Streams. *Big Data Research*, 9:98–106, 2017.

[206] Robert Keller, Lukas Häfner, Thomas Sachs, and Gilbert Fridgen. Scheduling Flexible Demand in Cloud Computing Spot Markets. *Business & Information Systems Engineering*, 62(1):25–39, 2020.

[207] Eamonn Keogh, Jessica Lin, and Ada Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 226–233, Washington, DC, USA, 2005. IEEE Computer Society.

[208] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 285–289, New York, NY, USA, 2000. ACM.

[209] Gil Keren and Björn W. Schuller. Convolutional RNN: an enhanced model for extracting features from sequential data. In *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pages 3412–3419. IEEE, 2016.

[210] Veena Khandelwal, Anand Kishore Chaturvedi, and Chandra Prakash Gupta. Amazon EC2 Spot Price Prediction Using Regression Random Forests. *IEEE Transactions on Cloud Computing*, 8(1):59–72, 2020.

[211] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S. Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2725–2732. ijcai.org, 2019.

[212] Franky Kin-Pong Chan, Ada Wai-chee Fu, and Clement Yu. Haar wavelets for efficient similarity search of time-series: With and without time warping. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):686–705, March 2003.

[213] Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10, 2019.

[214] Jerzy Kornowski. Linear prediction of aggregated seismic and seismoacoustic energy emitted from a mining longwall. *Acta Montana Ser. A*, 22(129):5–14, 2003.

[215] Marcin Kowalski, Dominik Ślęzak, Krzysztof Stencel, Przemysław Wiktor Pardel, Marek Grzegorowski, and Michal Kijowski. RDBMS Model for Scientific Articles Analytics, booktitle = Intelligent Tools for Building a Scientific Information Platform. volume 390 of *Studies in Computational Intelligence*, pages 49–60. Springer, 2012.

[216] Michal Kozielski, Marek Sikora, and Łukasz Wróbel. DISESOR - decision support system for mining industry. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015*, volume 5 of *Annals of Computer Science and Information Systems*, pages 67–74. IEEE, 2015.

[217] Bartosz Krawczyk, Leandro L. Minku, João Gama, Jerzy Stefanowski, and Michał Woźniak. Ensemble learning for data stream analysis: A survey. *Inf. Fusion*, 37:132–156, 2017.

[218] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[219] Marzena Kryszkiewicz. Rough set approach to incomplete information systems. *Information Sciences*, 112(1):39–49, 1998.

[220] Ludmila I. Kuncheva and Juan José Rodríguez Diez. On feature selection protocols for very low-sample-size data. *Pattern Recognit.*, 81:660–673, 2018.

[221] Karol Kurach and Krzysztof Pawłowski. Predicting dangerous seismic activity with recurrent neural networks. In M. Ganzha, L. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, volume 8 of *Annals of Computer Science and Information Systems*, pages 239–243. IEEE, 2016.

[222] Rosdyana Mangir Irawan Kusuma, Trang-Thi Ho, Wei-Chun Kao, Yu-Yen Ou, and Kai-Lung Hua. Using deep learning neural networks and candlestick chart representation to predict stock market, 2019.

[223] Petre Lameski, Eftim Zdravevski, Riste Mingov, and Andrea Kulakov. SVM Parameter Tuning with Grid Search and Its Impact on Reduction of Model Over-fitting. In Yiyu Yao, Qinghua Hu, Hong Yu, and Jerzy W. Grzymała-Busse, editors, *Proceedings of RSFDGrC 2015*, volume 9437 of *Lecture Notes in Computer Science*, pages 464–474. Springer, 2015.

[224] Gongmin Lan, Chenping Hou, Feiping Nie, Tingjin Luo, and Dongyun Yi. Robust Feature Selection via Simultaneous Sapped Norm and Sparse Regularizer Minimization. *Neurocomputing*, 283:228–240, 2018.

[225] Mattias Landfors, Philge Philip, Patrik Rydén, and Per Stenberg. Normalization of High Dimensional Genomics Data Where the Distribution of the Altered Variables Is Skewed. *PLOS ONE*, 6(11):1–11, 11 2011.

[226] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society, 2008.

[227] Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Commun. Surv. Tutorials*, 15(3):1192–1209, 2013.

[228] S. Lasocki. Probabilistic analysis of seismic hazard posed by mining induced events. In *Proceedings of sixth international symposium on rockburst and seismicity in mines*, pages 151–156. 2005.

[229] Cosmin Lazar, Jonatan Taminau, Stijn Meganck, David Steenhoff, Alain Coletta, Colin Molter, Virginie de Schaetzen, Robin Duque, Hugues Bersini, and Ann Nowé. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE ACM Trans. Comput. Biol. Bioinform.*, 9(4):1106–1119, 2012.

[230] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *ISCAS*, pages 253–256. IEEE, 2010.

[231] Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, and Bongki Moon. Parallel Data Processing with MapReduce: A Survey. *SIGMOD Record*, 40(4):11–20, 2012.

[232] Andrzej Leśniak and Zbigniew Isakow. Space-time clustering of seismic events and hazard assessment in the Zabrze-Bielszowice coal mine, Poland. *International Journal of Rock Mechanics and Mining Sciences*, 46(5):918–928, 2009.

[233] Alexandre L.M. Levada. Parametric PCA for unsupervised metric learning. *Pattern Recognition Letters*, 135:425 – 430, 2020.

[234] Chengwu Li and Dihao Ai. Automatic crack detection method for loaded coal in vibration failure process. *PLOS ONE*, 12(10):1–21, 10 2017.

[235] Mingsong Li, Linda Hinnov, and Lee Kump. Acycle: Time-series analysis software for paleoclimate research and education. *Computers & Geosciences*, 127:12 – 22, 2019.

[236] Ping Li, Jianyang Wu, and Lin Shang. Fast Approximate Attribute Reduction with MapReduce. In *Proc. of RSKT 2013*, pages 271–278.

[237] Chun-Cheng Lin, Der-Jiunn Deng, Chin-Hung Kuo, and Linnan Chen. Concept Drift Detection and Adaption in Big Imbalance Industrial IoT Data Using an Ensemble Learning Method of Offline Classifiers. *IEEE Access*, 7:56198–56207, 2019.

[238] Jessica Lin, Michail Vlachos, Eamonn Keogh, and Dimitrios Gunopulos. Iterative incremental clustering of time series. In Elisa Bertino, Stavros Christodoulakis, Dimitris Plexousakis, Vassilis Christophides, Manolis Koubarakis, Klemens Böhm, and Elena Ferrari, editors, *Advances in Database Technology - EDBT 2004: 9th International Conference on Extending Database Technology, Heraklion, Crete, Greece, March 14-18, 2004*, pages 106–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[239] Wei-Chao Lin and Chih-Fong Tsai. Missing value imputation: a review and analysis of the literature (2006-2017). *Artif. Intell. Rev.*, 53(2):1487–1509, 2020.

[240] Anjin Liu, Jie Lu, Feng Liu, and Guangquan Zhang. Accumulating regional density dissimilarity for concept drift detection in data streams. *Pattern Recognition*, 76:256 – 272, 2018.

[241] Bin Liu and Grigorios Tsoumakas. Dealing with class imbalance in classifier chains via random undersampling. *Knowledge-Based Systems*, 192:105292, 2020.

[242] Huan Liu and Hiroshi Motoda, editors. *Feature Extraction, Construction and Selection*. Springer, 1998.

[243] Huawen Liu, Xindong Wu, and Shichao Zhang. A New Supervised Feature Selection Method for Pattern Classification. *Computational Intelligence*, 30(2):342–361, 2014.

[244] Jianran Liu, Shiping Wang, and Wenyuan Yang. Sparse autoencoder for social image understanding. *Neurocomputing*, 369:122 – 133, 2019.

[245] Keyu Liu, Xibei Yang, Hualong Yu, Jusheng Mi, Pingxin Wang, and Xiangjian Chen. Rough set based semi-supervised feature selection via ensemble selector. *Knowl. Based Syst.*, 165:282–296, 2019.

[246] Xiaodong Liu and Witold Pedrycz. The development of fuzzy decision trees in the framework of axiomatic fuzzy set logic. *Applied Soft Computing*, 7(1):325 – 342, 2007.

[247] Yang Liu, Xinbo Gao, Quanxue Gao, Ling Shao, and Jungong Han. Adaptive robust principal component analysis. *Neural Networks*, 119:85 – 92, 2019.

[248] Chuan Luo, Tianrui Li, and Yiyu Yao. Dynamic probabilistic rough sets with incomplete data. *Inf. Sci.*, 417:39–54, 2017.

[249] Junfang Luo, Hamido Fujita, Yiyu Yao, and Keyun Qin. On modeling similarity and three-way decision under incomplete information in rough set theory. *Knowledge-Based Systems*, 191:105251, 2020.

[250] Jan Luts, Fabian Ojeda, Raf Van de Plas, Bart De Moor, Sabine Van Huffel, and Johan A.K. Suykens. A tutorial on support vector machine-based methods for classification problems in chemometrics. *Analytica Chimica Acta*, 665(2):129 – 145, 2010.

[251] Congcong Ma, Wenfeng Li, Jingjing Cao, Juan Du, Qimeng Li, and Raffaele Gravina. Adaptive sliding window based activity recognition for assisted livings. *Information Fusion*, 53:55 – 65, 2020.

[252] Majdi M. Mafarja and Seyedali Mirjalili. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, 260:302–312, 2017.

[253] Majdi M. Mafarja and Seyedali Mirjalili. Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. *Soft Comput.*, 23(15):6249–6265, 2019.

[254] Sebastián Maldonado and Julio López. Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification. *Appl. Soft Comput.*, 67:94–105, 2018.

[255] Ameya Malondkar, Roberto Corizzo, Iluju Kiringa, Michelangelo Ceci, and Nathalie Japkowicz. Spark-GHSOM: Growing Hierarchical Self-Organizing Map for large scale mixed attribute datasets. *Information Sciences*, 496:572 – 591, 2019.

[256] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[257] Christopher Mark. Coal Bursts in the Deep Longwall Mines of the United States. *International Journal of Coal Science & Technology*, 3(1):1–9, 2016.

[258] Alexina Jane Mason. *Bayesian methods for modelling non-random missing data mechanisms in longitudinal studies*. PhD thesis, Imperial College London, 2009.

[259] Sajee Mathew. Overview of Amazon Web Services, april 2017. Accessed: 2019-06-04.

[260] Michał Meina, Andrzej Janusz, Krzysztof Rykaczewski, Dominik Ślęzak, Bartosz Celmer, and Adam Krasuski. Tagging Firefighter Activities at the Emergency Scene: Summary of AAIA'15 Data Mining Competition at Knowledge Pit. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015*, volume 5 of *Annals of Computer Science and Information Systems*, pages 367–373. IEEE, 2015.

[261] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[262] Jan Kanty Milczek, Robert Bogucki, Jan Lasek, and Michał Tadeusiak. Early warning system for seismic events in coal mines using machine learning. In M. Ganzha, L. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, volume 8 of *Annals of Computer Science and Information Systems*, pages 213–220. IEEE, 2016.

[263] Fan Min, Qinghua Hu, and William Zhu. Feature Selection with Test Cost Constraint. *International Journal of Approximate Reasoning*, 55(1):167–179, 2014.

[264] T. Miranda, A. Gomes Correia, and L. Ribeiro e Sousa. Bayesian methodology for updating geomechanical parameters and uncertainty quantification. *International Journal of Rock Mechanics and Mining Sciences*, 46(7):1144 – 1153, 2009.

[265] Wojciech Moczulski, Piotr Przystałka, Marek Sikora, and Radosław Zimroz. Modern ICT and mechatronic systems in contemporary mining industry. In *Rough Sets - International Joint Conference, IJCRS 2016, Santiago de Chile, Chile, October 7-11, 2016, Proceedings*, pages 33–42, 2016.

[266] Muhidin Mohamed and Mourad Oussalah. SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management*, 56(4):1356–1372, 2019.

[267] Uwe Mönks, Helene Dörksen, Volker Lohweg, and Michael Hübner. Information Fusion of Conflicting Input Data. *Sensors*, 16(11):E1798, 2016.

[268] Ramon E. Moore, R. Baker Kearfott, and Michael J. Cloud. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 2009.

[269] Fabian Mörchen and Alfred Ultsch. Optimizing time series discretization for knowledge discovery. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 660–665, New York, NY, USA, 2005. ACM.

[270] Mikhail Ju. Moshkov, Marcin Piliszczuk, and Beata Zielosko. On Construction of Partial Reducts and Irreducible Partial Decision Rules. *Fundamenta Informaticae*, 75(1-4):357–374, 2007.

[271] Lijuan Mu and Yan Ji. Integrated coal mine safety monitoring system. In *Software Engineering and Knowledge Engineering: Theory and Practice: Selected papers from 2012 International Conference on Software Engineering, Knowledge Engineering and Information Engineering (SEKEIE 2012)*, pages 365–371. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[272] Francesco Mulargia, Philip B. Stark, and Robert J. Geller. Why is Probabilistic Seismic Hazard Analysis (PSHA) still used? *Physics of the Earth and Planetary Interiors*, 264:63 – 75, 2017.

[273] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[274] Hamid Nasiri, Saeed Nasehi, and Maziar Goudarzi. Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities. *J. Big Data*, 6:52, 2019.

[275] Hung Son Nguyen. Approximate Boolean Reasoning: Foundations and Applications in Data Mining. *Trans. Rough Sets*, 5:334–506, 2006.

[276] Hung Son Nguyen and Dominik Ślęzak. Approximate Reducts and Association Rules – Correspondence and Complexity Results. In Ning Zhong, Andrzej Skowron, and Setsuo Ohsuga, editors, *Proceedings of RSFDGrC 1999*, pages 137–145. Springer, 1999.

[277] Sinh Hoa Nguyen and Marcin S. Szczuka. Feature Selection in Decision Systems with Constraints. In *Proceedings of IJCRS 2016*, volume 9920 of *Lecture Notes in Computer Science*, pages 537–547. Springer, 2016.

[278] Son H. Nguyen and Andrzej Skowron. Quantization Of Real Value Attributes - Rough Set and Boolean Reasoning Approach. In *Proc. of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina, Sept 28 - Oct 1*, pages 34–37, 1995.

[279] Tuan Trung Nguyen and Andrzej Skowron. Rough-Granular Computing in Human-Centric Information Processing. In Andrzej Bargiela and Witold Pedrycz, editors, *Human-Centric Information Processing Through Granular Modelling*, volume 182 of *Studies in Computational Intelligence*, pages 1–30. Springer, 2009.

[280] Mark S. Nixon and Alberto S. Aguado. *Feature Extraction and Image Processing for Computer Vision*. Academic Press, fourth edition edition, 2020.

[281] Sarah Nogueira. *Quantifying the Stability of Feature Selection*. PhD thesis, University of Manchester, 2018.

[282] Sarah Nogueira, Konstantinos Sechidis, and Gavin Brown. On the stability of feature selection algorithms. *J. Mach. Learn. Res.*, 18:174:1–174:54, 2017.

[283] Raul-Jose Palma-Mendoza, Daniel Rodríguez, and Luis de Marcos. Distributed ReliefF-based feature selection in Spark. *Knowl. Inf. Syst.*, 57(1):1–20, 2018.

[284] Niki Parmar, Prajit Ramachandran, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 68–80, 2019.

[285] Zdzisław Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*, volume 9 of *System Theory, Knowledge Engineering and Problem Solving*. Kluwer, 1991.

[286] Zdzisław Pawlak and Andrzej Skowron. Advances in the Dempster-Shafer Theory of Evidence. chapter Rough Membership Functions, pages 251–271. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[287] Zdzisław Pawlak and Andrzej Skowron. Rough sets: Some extensions. *Information Sciences*, 177(1):28–40, 2007.

[288] Zdzisław Pawlak and Andrzej Skowron. Rudiments of Rough Sets. *Information Sciences*, 177(1):3–27, 2007.

[289] Krzysztof Pawłowski and Karol Kurach. Detecting Methane Outbreaks from Time Series Data with Deep Neural Networks. In Yiyu Yao, Qinghua Hu, Hong Yu, and Jerzy W. Grzymała-Busse, editors, *Proceedings of RSFDGrC 2015*, volume 9437 of *Lecture Notes in Computer Science*, pages 494–500. Springer, 2015.

[290] Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

[291] Witold Pedrycz. Interpretation of clusters in the framework of shadowed sets. *Pattern Recognition Letters*, 26(15):2439–2449, 2005.

[292] Witold Pedrycz. *Granular Computing: Analysis and Design of Intelligent Systems*. CRC Press, 2013.

[293] Witold Pedrycz. Granular computing for data analytics: a manifesto of human-centric computing. *IEEE CAA J. Autom. Sinica*, 5(6):1025–1034, 2018.

[294] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[295] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

[296] Jose Alberto Perez-Benitez and Linilson Rodrigues Padovese. A system for classification of time-series data from industrial non-destructive device. *Engineering Applications of Artificial Intelligence*, 26(3):974–983, 2013.

[297] Lars-Erik Persson, Natasha Samko, and Peter Wall. Quasi-monotone weight functions and their characteristics and applications. *Mathematical Inequalities & Applications*, 15:685–705, 07 2012.

[298] Łukasz Podlodowski. Utilizing an ensemble of SVMs with GMM voting-based mechanism in predicting dangerous seismic events in active coal mines. In M. Ganzha, L. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, volume 8 of *Annals of Computer Science and Information Systems*, pages 235–238. IEEE, 2016.

[299] Robi Polikar, Joseph DePasquale, Hussein Syed Mohammed, Gavin Brown, and Ludmilla I. Kuncheva. Learn++.MF: A Random Subspace Approach for the Missing Feature Problem. *Pattern Recognition*, 43(11):3817–3832, 2010.

[300] Vasco Ponciano, Ivan Miguel Pires, Fernando Reinaldo Ribeiro, María Vanessa Villasana, Rute Crisóstomo, Maria Cristina Canavarro Teixeira, and Eftim Zdravevski. Mobile computing technologies for health and mobility assessment: Research design and results of the timed up and go test in older adults. *Sensors*, 20(12):3481, 2020.

[301] Jean-Christophe Popieul, Pierre Loslever, Alexis Todoskoff, Philippe Simon, and Matthias Rotting. Multivariate analysis of human behavior data using fuzzy windowing: Example with driver-car-environment system. *Engineering Applications of Artificial Intelligence*, 25(5):989–996, 2012.

[302] Kedar Potdar, Taher Pardawala, and Chinmay Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175:7–9, 10 2017.

[303] Piotr Przystałka and Marek Sikora, editors. *Zintegrowany, szkieletowy system wspmagania decyzji dla systemów monitorowania procesów, urządzeń i zagrożeń.* Monograficzna Seria Wydawnicza Instyututu Technik Innowacyjnych EMAG, 2017.

[304] J. Qian, Duoqian Miao, Z.H. Zhang, and W. Li. Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation. *International Journal of Approximate Reasoning*, 52(2):212–230, 2011. Philosophy of Probability.

[305] Jimin Qian, Nam Phuong Nguyen, Yutaka Oya, Gota Kikugawa, Tomonaga Okabe, Yue Huang, and Fumio S. Ohuchi. Introducing self-organized maps (SOM) as a visualization tool for materials research and education. *Results in Materials*, 4, 2019.

[306] Jin Qian, Chuangyin Dang, Xiaodong Yue, and Nan Zhang. Attribute reduction for sequential three-way decisions under dynamic granulation. *International Journal of Approximate Reasoning*, 85:196 – 216, 2017.

[307] Jin Qian, Ping Lv, Xiaodong Yue, Caihui Liu, and Zhengjun Jing. Hierarchical Attribute Reduction Algorithms for Big Data Using MapReduce. *Knowledge-Based Systems*, 73(C):18–31, 2015.

[308] J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[309] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[310] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 2019.

[311] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[312] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. *ACM Trans. Knowl. Discov. Data*, 7(3):10:1–10:31, 2013.

[313] Sergio Ramírez-Gallego, Iago Lastra, David Martínez-Rego, Verónica Bolón-Canedo, José Manuel Benítez, Francisco Herrera, and Amparo Alonso-Betanzos. Fast-mrmr: Fast minimum redundancy maximum relevance algorithm for high-dimensional big data. *Int. J. Intell. Syst.*, 32:134–152, 2017.

[314] Yingli Ran, Yishuo Shi, and Zhao Zhang. Local Ratio Method on Partial Set Multi-cover. *Journal of Combinatorial Optimization*, 34(1):302–313, 2017.

[315] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. Training complex models with multi-task weak supervision. In *AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4763–4771. AAAI Press, 2019.

[316] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains: A review and perspectives. *CoRR*, abs/1912.13405, 2019.

[317] Jesse Read, Antti Puurula, and Albert Bifet. Multi-label classification with meta-labels. In Ravi Kumar, Hannu Toivonen, Jian Pei, Joshua Zhexue Huang, and Xindong Wu, editors, *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*, pages 941–946. IEEE Computer Society, 2014.

[318] Muhammad Habib Rehman, Victor Chang, Aisha Batool, and Teh Ying Wah. Big Data Reduction Framework for Value Creation in Sustainable Enterprises. *International Journal of Information Management*, 36(6):917–928, 2016.

[319] Lala Septem Riza, Andrzej Janusz, Christoph Bergmeir, Chris Cornelis, Francisco Herrera, Dominik Ślęzak, and José Manuel Benítez. Implementing Algorithms of Rough Set Theory and Fuzzy Rough Set Theory in the R Package 'RoughSets'. *Information Sciences*, 287:68–89, 2014.

[320] Henriette Röger and Ruben Mayer. A comprehensive survey on parallelization and elasticity in stream processing. *ACM Comput. Surv.*, 52(2), 2019.

[321] Joshua Rosen, Neoklis Polyzotis, Vinayak R. Borkar, Yingyi Bu, Michael J. Carey, Markus Weimer, Tyson Condie, and Raghu Ramakrishnan. Iterative MapReduce for Large Scale Machine Learning. *CoRR*, abs/1303.3517, 2013.

[322] Amitava Roy and Sankar K. Pal. Fuzzy discretization of feature space for a rough set classifier. *Pattern Recogn. Lett.*, 24(6):895–902, March 2003.

[323] Debaditya Roy, K. Sri Rama Murty, and C. Krishna Mohan. Feature Selection using Deep Neural Networks. In *Proceedings of IJCNN 2015*, pages 1–6, 2015.

[324] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[325] Cynthia Rudin. Please stop explaining black box models for high stakes decisions. *CoRR*, abs/1811.10154, 2018.

[326] Ernestina Menasalvas Ruiz, Juan Manuel Tuñas, Guzmán Bermejo, Consuelo Gonzalo-Martín, Alejandro Rodríguez González, Massimiliano Zanin, Cristina Gonzalez de Pedro, Marta Mendez, Olga Zaretskaia, Jesús Rey, Consuelo Parejo, Juan Luis Cruz Bermudez, and Mariano Provencio. Profiling lung cancer patients using electronic health records. *J. Medical Systems*, 42(7):126:1–126:10, 2018.

[327] Dymitr Ruta and Ling Cen. Self-Organized Predictor of Methane Concentration Warnings in Coal Mines. In Yiyu Yao, Qinghua Hu, Hong Yu, and Jerzy W. Grzymała-Busse, editors, *Proceedings of RSFDGrC 2015*, volume 9437 of *Lecture Notes in Computer Science*, pages 485–493. Springer, 2015.

[328] Jarosław Rzeszótko and Sinh Hoa Nguyen. Machine Learning for Traffic Prediction. *Fundamenta Informaticae*, 119(3-4):407–420, 2012.

[329] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 313–325, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[330] Syed Moshfeq Salaken, Abbas Khosravi, Thanh Nguyen, and Saeid Nahavandi. Seeded transfer learning for regression problems with deep learning. *Expert Syst. Appl.*, 115:565–577, 2019.

[331] Sunita Sarawagi, Shiby Thomas, and Rakesh Agrawal. Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications. *Data Mining and Knowledge Discovery*, 4(2-3):89–125, 2000.

[332] Maximilian Schaefer and Matthias Eikermann. Contact-free respiratory monitoring using bed-wheel sensors: A valid respiratory monitoring technique with significant potential impact on public health. *Journal of Applied Physiology*, 126, 03 2019.

[333] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, pages 357–360, New York, NY, USA, 2007. ACM.

[334] Azlyna Senawi, Hua-Liang Wei, and Stephen A. Billings. A New Maximum Relevance-Minimum Multicollinearity (MRmMC) Method for Feature Selection and Ranking. *Pattern Recognition*, 67:47–61, 2017.

[335] Mikel Sesma-Sara, Radko Mesiar, and Humberto Bustince. Weak and directional monotonicity of functions on Riesz spaces to fuse uncertain data. *Fuzzy Sets and Systems*, 386:145–160, 2020. Aggregation Operations.

[336] Omer Berat Sezer and Ahmet Murat Ozbayoglu. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70:525–538, 2018.

[337] Dev Shah, Haruna Isah, and Farhana Zulkernine. Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 7(2), 2019.

[338] Jasmit SureshKumar Shah. *Novel statistical approaches for missing values in truncated high-dimensional metabolomics data with a detection threshold.* PhD thesis, University of Louisville, 2017.

[339] Radwa El Shawi, Sherif Sakr, Domenico Talia, and Paolo Trunfio. Big Data Systems Meet Machine Learning Challenges: Towards Big Data Science as a Service. *Big Data Research*, 14:1–11, 2018.

[340] Yan-Hong She, Zhuo-Hao Qian, Xiao-Li He, Jun-Tao Wang, Ting Qian, and Wen-Li Zheng. On generalization reducts in multi-scale decision tables. *Information Sciences*, 555:104–124, 2021.

[341] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, and Mohammad Ali Zare Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognit.*, 64:141–158, 2017.

[342] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, and Mohammad Ali Zare Chahooki. A robust graph-based semi-supervised sparse feature selection method. *Inf. Sci.*, 531:13–30, 2020.

[343] Seyed Amin Seyfi Shishavan, Fatma Kutlu Gündogdu, Elmira Farrokhizadeh, Yaser Donyatalab, and Cengiz Kahraman. Novel similarity measures in spherical fuzzy environment and their applications. *Eng. Appl. Artif. Intell.*, 94:103837, 2020.

[344] Iftikhar U. Sikder and Toshinori Munakata. Application of rough set and decision tree for characterization of premonitory factors of low seismic activity. *Expert Systems with Applications*, 36(1):102–110, 2009.

[345] Marek Sikora and Beata Sikora. Improving prediction models applied in systems monitoring natural hazards and machinery. *International Journal of Applied Mathematics and Computer Science*, 22(2):477–491, 2012.

[346] Pritpal Singh and Gaurav Dhiman. A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. *Journal of Computational Science*, 27:370 – 385, 2018.

[347] Andrzej Skowron and Soma Dutta. Rough sets: past, present, and future. *Nat. Comput.*, 17(4):855–876, 2018.

[348] Andrzej Skowron, Andrzej Jankowski, and Soma Dutta. Interactive granular computing. *Granular Computing*, 1(2):95–113, 2016.

[349] Andrzej Skowron and Cecylia Rauszer. The discernibility matrices and functions in information systems. In Roman Słowiński, editor, *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, volume 11 of *Theory and Decision Library*, pages 331–362. Springer, 1992.

[350] Andrzej Skowron and Piotr Wasilewski. Interactive information systems: Toward perception based computing. *Theor. Comput. Sci.*, 454:240–260, 2012.

[351] Dominik Ślęzak. Normalized Decision Functions and Measures for Inconsistent Decision Tables Analysis. *Fundamenta Informaticae*, 44(3):291–319, 2000.

[352] Dominik Ślęzak. Approximate Entropy Reducts. *Fundamenta Informaticae*, 53(3-4):365–390, 2002.

[353] Dominik Ślęzak. Rough Sets and Functional Dependencies in Data: Foundations of Association Reducts. *Transactions on Computational Science*, 5:182–205, 2009.

[354] Dominik Ślęzak. Compound Analytics of Compound Data within RDBMS Framework - Infobright's Perspective. In Tai-Hoon Kim, Young-Hoon Lee, Byeong Ho Kang, and Dominik Slezak, editors, *Future Generation Information Technology - Second International Conference, FGIT 2010, Jeju Island, Korea, December 13-15, 2010. Proceedings*, volume 6485 of *Lecture Notes in Computer Science*, pages 39–40. Springer, 2010.

[355] Dominik Ślęzak, Rick Glick, Paweł Betliński, and Piotr Synak. A new approximate query engine based on intelligent capture and fast transformations of granulated data summaries. *J. Intell. Inf. Syst.*, 50(2):385–414, 2018.

[356] Dominik Ślęzak, Marek Grzegorowski, Andrzej Janusz, Michał Kozielski, Sinh Hoa Nguyen, Marek Sikora, Sebastian Stawicki, and Łukasz Wróbel. A Framework for Learning and Embedding Multi-Sensor Forecasting Models into a Decision Support System: A Case Study of Methane Concentration in Coal Mines. *Information Sciences*, 451-452:112–133, 2018.

[357] Dominik Ślęzak, Marek Grzegorowski, Andrzej Janusz, and Sebastian Stawicki. Interactive Data Exploration with Infolattices. Abstract Materials of BAFI 2015.

[358] Dominik Ślęzak, Marek Grzegorowski, Andrzej Janusz, and Sebastian Stawicki. Toward Interactive Attribute Selection with Infolattices - A Position Paper. In *IJCRS (2)*, volume 10314 of *Lecture Notes in Computer Science*, pages 526–539. Springer, 2017.

[359] Dominik Ślęzak and Andrzej Janusz. Ensembles of Bireducts: Towards Robust Classification and Simple Representation. In Tai-Hoon Kim, Hojjat Adeli, Dominik Ślęzak, Frode Eika Sandnes, Xiaofeng Song, Kyo-Il Chung, and Kirk P. Arnett, editors, *Future Generation Information Technology - Third International Conference, FGIT 2011 in Conjunction with GDC 2011, Jeju Island, Korea, December 8-10, 2011. Proceedings*, volume 7105 of *Lecture Notes in Computer Science*, pages 64–77. Springer, 2011.

[360] Dominik Ślęzak and Sebastian Stawicki. The Problem of Finding the Simplest Classifier Ensemble is NP-Hard - A Rough-Set-Inspired Formulation Based on Decision Bireducts. In Rafael Bello, Duoqian Miao, Rafael Falcon, Michinori Nakata, Alejandro Rosete, and Davide Ciucci, editors, *Rough Sets - International Joint Conference, IJCRS 2020, Havana, Cuba, June 29 - July 3, 2020, Proceedings*, volume 12179 of *Lecture Notes in Computer Science*, pages 204–212. Springer, 2020.

[361] Dominik Ślęzak and Sebastian Widz. Evolutionary inspired optimization of feature subset ensembles. In Hideyuki Takagi, Ajith Abraham, Mario Köppen, Kaori Yoshida, and André C. P. L. F. de Carvalho, editors, *Second World Congress on Nature & Biologically Inspired Computing, NaBIC 2010, 15-17 December 2010, Kitakyushu, Japan*, pages 437–442. IEEE, 2010.

[362] Melanie Smuk. *Missing Data Methodology: Sensitivity analysis after multiple imputation.* PhD thesis, University of London, 2015.

[363] Parinaz Sobhani, Herna Viktor, and Stan Matwin. Learning from imbalanced data using ensemble methods and cluster-based undersampling. In Annalisa Appice, Michelangelo Ceci, Corrado Loglisci, Giuseppe Manco, Elio Masciari, and Zbigniew W. Ras, editors, *New Frontiers in Mining Complex Patterns*, pages 69–83, Cham, 2015. Springer International Publishing.

[364] Le Hoang Son. Dealing with the new user cold-start problem in recommender systems: A comparative review. *Information Systems*, 58:87–104, 2016.

[365] C. O. S. Sorzano, J. Vargas, and A. Pascual Montano. A survey of dimensionality reduction techniques, 2014.

[366] Somayajulu G. Sripada, Ehud Reiter, Jim Hunter, Jin Yu, and Ian P. Davy. Modelling the task of summarising time series data using ka techniques. In Ann Macintosh, Mike Moulton, and Alun Preece, editors, *Applications and Innovations in Intelligent Systems IX: Proceedings of ES2001, the Twenty-first SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence, Cambridge, December 2001*, pages 183–196. Springer London, London, 2002.

[367] Sebastian Stawicki, Dominik Ślęzak, Andrzej Janusz, and Sebastian Widz. Decision Bireducts and Decision Reducts – A Comparison. *International Journal of Approximate Reasoning*, 84:75–109, 2017.

[368] Hervé Stoppiglia, Gérard Dreyfus, Rémi Dubois, and Yacine Oussar. Ranking a random feature for variable and feature selection. *J. Mach. Learn. Res.*, 3:1399–1414, 2003.

[369] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the Importance of Initialization and Momentum in Deep Learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages 1139–1147. JMLR.org, 2013.

[370] Roman W. Świniarski and Andrzej Skowron. Rough Set Methods in Feature Selection and Recognition. *Pattern Recognition Letters*, 24(6):833–849, 2003.

[371] Marcin S. Szczuka and Dominik Ślęzak. How Deep Data Becomes Big Data. In *Proc. of IFSA/NAFIPS 2013*, pages 579–584.

[372] Marcin S. Szczuka and Piotr Wojdyłło. Neuro-Wavelet Classifiers for EEG Signals Based on Rough Set Methods. *Neurocomputing*, 36(1-4):103–122, 2001.

[373] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[374] Jerffeson Teixeira de Souza, Stan Matwin, and Nathalie Japkowicz. Parallelizing Feature Selection. *Algorithmica*, 45(3):433–456, 2006.

[375] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics, 2019.

[376] Thanh N. Tran, Nelson Lee Afanador, Lutgarde M.C. Buydens, and Lionel Blanchet. Interpretation of variable importance in Partial Least Squares with Significance Multivariate Correlation (SMC). *Chemometrics and Intelligent Laboratory Systems*, 138:153–160, 2014.

[377] Isaac Triguero, Daniel Peralta, Jaume Bacardit, Salvador García, and Francisco Herrera. MRPR: A MapReduce Solution for Prototype Reduction in Big Data Classification. *Neurocomputing*, 150:331–345, 2015.

[378] Chih-Fong Tsai and Yu-Chi Chen. The optimal combination of feature selection and data discretization: An empirical study. *Information Sciences*, 505:282 – 293, 2019.

[379] Nikolaos L. Tsakiridis, Themistoklis Diamantopoulos, Andreas L. Symeonidis, John B. Theocharis, Athanasios Iossifides, Periklis Chatzimisios, George Pratos, and Dimitris Kouvas. Versatile internet of things for agriculture: An explainable ai approach. In Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis, editors, *Artificial Intelligence Applications and Innovations*, pages 180–191, Cham, 2020. Springer International Publishing.

[380] Ryan J. Urbanowicz, Melissa Meeker, William G. La Cava, Randal S. Olson, and Jason H. Moore. Relief-based feature selection: Introduction and review. *J. Biomed. Informatics*, 85:189–203, 2018.

[381] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative review. Tilburg University Technical Report, TiCC-TR 2009, 2009.

[382] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[383] Jorge R. Vergara and Pablo A. Estévez. A review of feature selection methods based on mutual information. *Neural Comput. Appl.*, 24(1):175–186, 2014.

[384] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA, 2008. ACM.

[385] Jana Špirková, Gleb Beliakov, Humberto Bustince, and Javier Fernandez. Mixture functions and their monotonicity. *Information Sciences*, 481:520–549, 2019.

[386] Dominik Wachla and Wojciech A. Moczulski. Identification of dynamic diagnostic models with the use of methodology of knowledge discovery in databases. *Engineering Applications of Artificial Intelligence*, 20(5):699–707, 2007.

[387] Hai Wang, Zeshui Xu, Hamido Fujita, and Shousheng Liu. Towards felicitous decision making: An overview on challenges and trends of Big Data. *Information Sciences*, 367-368:747–765, nov 2016.

[388] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, May 2013.

[389] Hongzhi Wang, Mohamed Jaward Bah, and Mohamed Hammad. Progress in outlier detection techniques: A survey. *IEEE Access*, 7:107964–108000, 2019.

[390] Linda Wang, Zhong Qiu Lin, and Alexander Wong. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific reports*, 10, 2020.

[391] Lipo Wang, Yaoli Wang, and Qing Chang. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods*, 111:21 – 31, 2016. Big Data Bioinformatics.

[392] Xiaoguang Wang, Xuan Liu, Nathalie Japkowicz, and Stan Matwin. Resampling and Cost-Sensitive Methods for Imbalanced Multi-instance Learning. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 808–816, 2013.

[393] Sebastian Widz and Dominik Ślęzak. Granular Attribute Selection: A Case Study of Rough Set Approach to MRI Segmentation. In *Proc. of PReMI 2013*, pages 47–52.

[394] Alicja Wieczorkowska, Jakub Wróblewski, Piotr Synak, and Dominik Ślęzak. Application of Temporal Descriptors to Musical Instrument Sound Recognition. *Journal of Intelligent Information Systems*, 21(1):71–93, 2003.

[395] Andrzej Wójtowicz. *Ensemble classification of incomplete data – a non-imputation approach with an application in ovarian tumour diagnosis support.* PhD thesis, University in Poznań, 2017.

[396] Andrzej Wójtowicz, Patryk Żywica, Anna Stachowiak, and Krzysztof Dyczkowski. Solving the problem of incomplete data in medical diagnosis via interval modeling. *Appl. Soft Comput.*, 47:424–437, 2016.

[397] Jakub Wróblewski. Ensembles of Classifiers Based on Approximate Reducts. *Fundamenta Informaticae*, 47(3-4):351–360, 2001.

[398] Jakub Wróblewski and Sebastian Stawicki. SQL-based KDD with Infobright's RDBMS: Attributes, Reducts, Trees. In *Proceedings of RSEISP 2014*, pages 28–41, 2014.

[399] Caesar Wu, Rajkumar Buyya, and Kotagiri Ramamohanarao. Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges. *ACM Comput. Surv.*, 52(6):108:1–108:36, 2020.

[400] Xiaohu Wu, Francesco De Pellegrini, Guanyu Gao, and Giuliano Casale. A Framework for Allocating Server Time to Spot and On-Demand Services in Cloud Computing. *TOMPECS*, 4(4):20:1–20:31, 2019.

[401] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 Algorithms in Data Mining. *Knowl. Inf. Syst.*, 14(1):1–37, December 2007.

[402] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 753–763, New York, NY, USA, 2020. Association for Computing Machinery.

[403] J. Xie, J. Wu, and Q. Qian. Feature Selection Algorithm Based on Association Rules Mining Method. In *Proc. of ICIS 2009*, pages 357–362.

[404] Eleftherios Spyromitros Xioufis, Myra Spiliopoulou, Grigorios Tsoumakas, and Ioannis Vlahavas. Dealing with concept drift and class imbalance in multi-label stream classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two*, IJCAI'11, pages 1583–1588. AAAI Press, 2011.

[405] Eleftherios Spyromitros Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis P. Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Mach. Learn.*, 104(1):55–98, 2016.

[406] Bing Xue, Mengjie Zhang, Will N. Browne, and Xin Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Trans. Evol. Comput.*, 20(4):606–626, 2016.

[407] Honggang Yang, Huibin Lin, and Kang Ding. Sliding window denoising k-singular value decomposition and its application on rolling bearing impact fault diagnosis. *Journal of Sound and Vibration*, 421:205 – 219, 2018.

[408] Yiming Yang and Siddharth Gopal. Multilabel classification with meta-level features in a learning-to-rank framework. *Machine Learning*, 88(1-2):47–68, 2012.

[409] Ying Yang and Geoffrey I. Webb. Discretization for naive-bayes learning: Managing discretization bias and variance. *Mach. Learn.*, 74(1):39–74, January 2009.

[410] Ying Yang, Geoffrey I. Webb, and Xindong Wu. Discretization methods. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook.*, pages 113–130. Springer, 2005.

[411] Yiyu Yao. Three-way decision and granular computing. *Int. J. Approx. Reasoning*, 103:107–123, 2018.

[412] Yiyu Yao, Yan Zhao, and Jue Wang. On Reduct Construction Algorithms. *Transactions on Computational Science*, 2:100–117, 2008.

[413] Yiyu Yao and Ning Zhong. Granular Computing. In Benjamin W. Wah, editor, *Wiley Encyclopedia of Computer Science and Engineering*. Wiley, 2008.

[414] Jiateng Yin and Wentian Zhao. Fault diagnosis network design for vehicle on-board equipments of high-speed railway: A deep learning approach. *Engineering Applications of Artificial Intelligence*, 56:250–259, 2016.

[415] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[416] Lotfi A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy sets and systems*, 90(2):111–127, 1997.

[417] Lotfi A. Zadeh. From Computing with Numbers to Computing with Words - From Manipulation of Measurements to Manipulation of Perceptions. In Behnam Azvine, Nader Azarmi, and Detlef D. Nauck, editors, *Intelligent Systems and Soft Computing: Prospects, Tools and Applications*, volume 1804 of *Lecture Notes in Computer Science*, pages 3–40. Springer, 2000.

[418] Adam Zagorecki. Prediction of Methane Outbreaks in Coal Mines from Multivariate Time Series Using Random Forest. In Yiyu Yao, Qinghua Hu, Hong Yu, and Jerzy W. Grzymała-Busse, editors, *Proceedings of RSFDGrC 2015*, volume 9437 of *Lecture Notes in Computer Science*, pages 494–500. Springer, 2015.

[419] Adam Zagorecki. A versatile approach to classification of multivariate time series data. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Lódz, Poland, September 13-16, 2015*, pages 407–410. IEEE, 2015.

[420] Eftim Zdravevski, Petre Lameski, Ace Dimitrievski, Marek Grzegorowski, and Cas Apanowicz. Cluster-size optimization within a cloud-based ETL framework for Big Data. In *2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019*, pages 3754–3763. IEEE, 2019.

[421] Eftim Zdravevski, Petre Lameski, and Andrea Kulakov. Automatic feature engineering for prediction of dangerous seismic activities in coal mines. In M. Ganzha, L. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, volume 8 of *Annals of Computer Science and Information Systems*, pages 245–248. IEEE, 2016.

[422] Eftim Zdravevski, Petre Lameski, Riste Mingov, Andrea Kulakov, and Dejan Gjorgjevikj. Robust Histogram-based Feature Engineering of Time Series Data. In *Proc. of FedCSIS 2015*, pages 381–388.

[423] Eftim Zdravevski, Petre Lameski, Vladimir Trajkovik, Andrea Kulakov, Ivan Chorbev, Rossitza Goleva, Nuno Pombo, and Nuno M. Garcia. Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering. *IEEE Access*, 5:5262–5280, 2017.

[424] Shichao Zhang, Chengqi Zhang, and Qiang Yang. Data preparation for data mining. *Applied Artificial Intelligence*, 17(5-6):375–381, 2003.

[425] Xianli Zhang, Buyue Qian, Shilei Cao, Yang Li, Hang Chen, Yefeng Zheng, and Ian Davidson. Inprem: An interpretable and trustworthy predictive model for healthcare. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 450–460, New York, NY, USA, 2020. Association for Computing Machinery.

[426] Yuanjian Zhang, Duoqian Miao, Witold Pedrycz, Tianna Zhao, Jianfeng Xu, and Ying Yu. Granular structure-based incremental updating for multi-label classification. *Knowl. Based Syst.*, 189, 2020.

[427] Xue Rong Zhao and Yiyu Yao. Three-way fuzzy partitions defined by shadowed sets. *Inf. Sci.*, 497:23–37, 2019.

[428] Yuxuan Zhao and Madeleine Udell. Missing value imputation for mixed data via gaussian copula. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 636–646, New York, NY, USA, 2020. Association for Computing Machinery.

[429] Zheng Zhao, Ruiwen Zhang, James Cox, David Duling, and Warren Sarle. Massively Parallel Feature Selection: An Approach Based on Variance Preservation. *Machine Learning*, 92(1):195–220, 2013.

[430] Wei Zheng, Xiaofeng Zhu, Guoqiu Wen, Yonghua Zhu, Hao Yu, and Jiangzhang Gan. Unsupervised feature selection by self-paced learning regularization. *Pattern Recognition Letters*, 132:4 – 11, 2020.

[431] Pengfei Zhu, Wencheng Zhu, Qinghua Hu, Changqing Zhang, and Wangmeng Zuo. Subspace clustering guided unsupervised feature selection. *Pattern Recognition*, 66:364 – 374, 2017.

[432] Wei Zong, Yang-Wai Chow, and Willy Susilo. Interactive three-dimensional visualization of network intrusion detection data for machine learning. *Future Gener. Comput. Syst.*, 102:292–306, 2020.

Appendix A

Data Insights

A.1 Methane Data

The appendix presents selected, more in-depth insights into the methane-related data set, reffed to in Section 5.1.2. The data correspond to a mining period between March 2, 2014, and June 16, 2014, and is collected from selected 28 (out of thousands) sensors located in an active Polish coal mine, which were located in the vicinity of the coal extraction area. The data has been made available on KnowlegePit platform for the purpose of organizing IJCRS'15 Data Challenge: Mining Data from Coal Mines [181].

Table A.1: Sensors related to methane data.

Sensor	Type	Unit	Type	Additional Info
AN311	Anemometer	m/s	alarming	Threshold A: none, Threshold B: <= 0.3 m/s
AN422	Anemometer	m/s	switching off	Threshold A: <= 1.1 m/s, Threshold B: <= 1.3 m/s
AN423	Anemometer	m/s	switching off	Threshold A: <= 1.0 m/s, Threshold B: <= 1.2 m/s
TP1721	Thermometer	$^{\circ}C$	registering	Tri-constituent sensor THP2/93
RH1722	Humidity	%RH	registering	Tri-constituent sensor THP2/93
BA1723	Barometer	hPa	registering	Tri-constituent sensor THP2/93
TP1711	Thermometer	$^{\circ}C$	registering	Tri-constituent sensor THP2/94
RH1712	Humidity	%RH	registering	Tri-constituent sensor THP2/94
BA1713	Barometer	hPa	registering	Tri-constituent sensor THP2/94
MM252	Methanometer	$\%CH_4$	switching off	Threshold A: 2.0%, Threshold B: 1.5%
MM261	Methanometer	$\%CH_4$	switching off	Threshold A: 1.5%, Threshold B: 1.0%
MM262	Methanometer	$\%CH_4$	switching off	Threshold A: 1.0%, Threshold B: 0.6%
MM263	Methanometer	$\%CH_4$	switching off	Threshold A: 1.5%, Threshold B: 1.0%
MM264	Methanometer	$\%CH_4$	switching off	Threshold A: 1.5%, Threshold B: 1.0%
MM256	Methanometer	$\%CH_4$	switching off	Threshold A: 1.5%,Threshold B: 1.0%
MM211	Methanometer	$\%CH_4$	switching off	Threshold A: 2.0%, Threshold B: 1.5%
CM861	Methanometer	$\%CH_4$	registering	Measures high concentrations of methane
CR863	Pressure difference	Pa	registering	Sensor is placed on the demethanisation orifice
P_864	Barometer	kPa	registering	Pressure inside the pipeline for methane drainage
TC862	Temperature	$^{\circ}C$	registering	Temperature inside the pipeline for methane drainage
WM868	Methane expense	m^3/min	registering	Methane expense calculated by CM, CR, P, TC
AMP1	Ammeter	A	registering	Current in the motor in the left arm of the shearer
AMP2	Ammeter	A	registering	Current in the motor in the right arm of the shearer
DMP3	Ammeter	A	registering	Current in the motor in the left tractor of the shearer
DMP4	Ammeter	A	registering	Current in the motor in the right tractor of the shearer
AMP5	Ammeter	A	registering	Current in the hydraulic pump motor of the shearer
F_SIDE	Drive direction	left, right	registering	The driving direction of the shearer
V	Shearer speed	Hz	registering	Work frequency, 100Hz means ca 20 m/min

In Table A.1, three groups of sensors are presented (groups are separated with horizontal lines). The first group is responsible for the monitoring of the mine atmosphere, the second group monitors the methane drainage flange, the third group monitors the operating status of a longwall shearer. The fifth column provides additional information about security thresholds assigned to the selected sensors. After crossing the threshold A the "switching off" sensors cut off the electricity supply. After crossing the threshold B both the "alarming" and "switching off" sensors display a predefined warning message. All of sensor recordings are collected and stored for the purpose of the further analysis. A detailed location of all sensors, as well as a workplace of a longwall shearer, on a fragment of the coal mine plan, is shown in Figure 5.2, in Section 5.1.

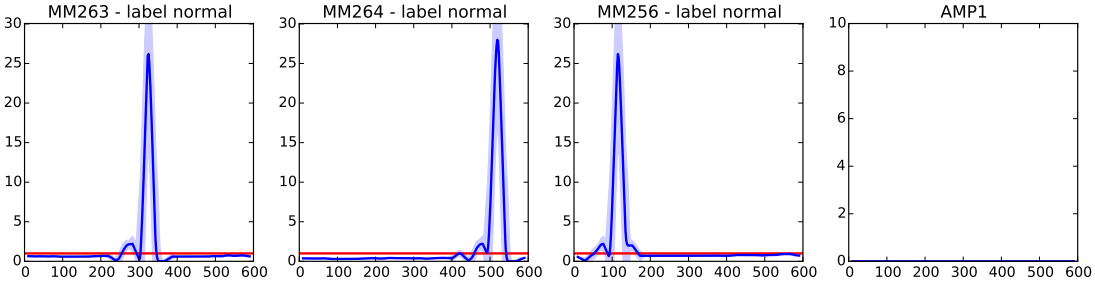


Figure A.1: Examples of outliers in methane concentration time series.

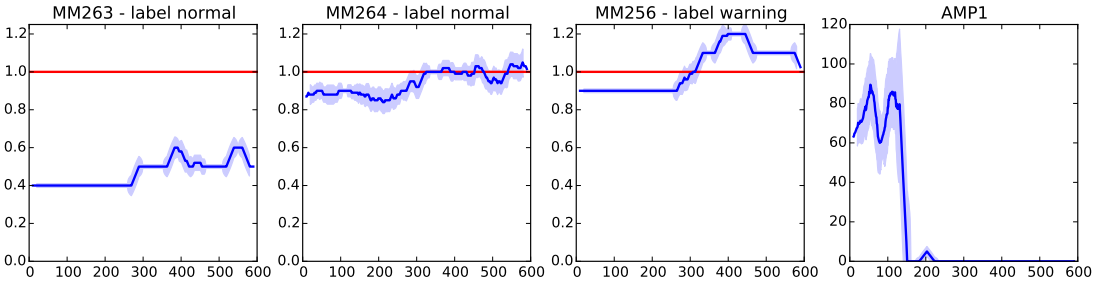


Figure A.2: Methane indications oscillating near the warning threshold. On the rightmost plot the current cut off after exceeding the methane concentration threshold.

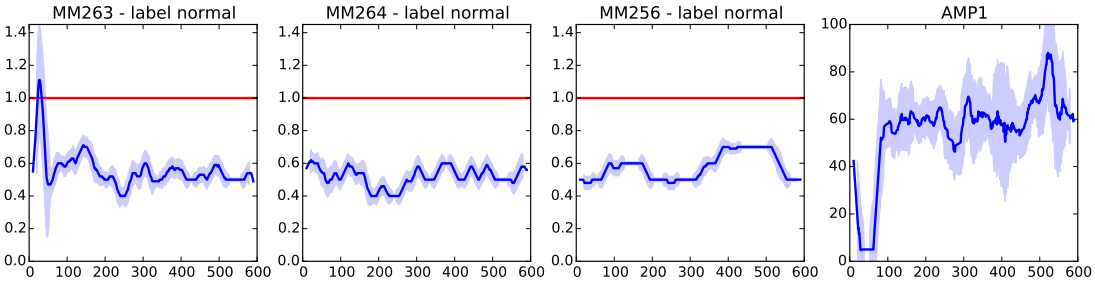


Figure A.3: Relatively small dynamics of changes in methane concentration for three methane detectors. On the rightmost plot we can observe current consumption of the cutter loader that corresponds to ongoing coal mining.

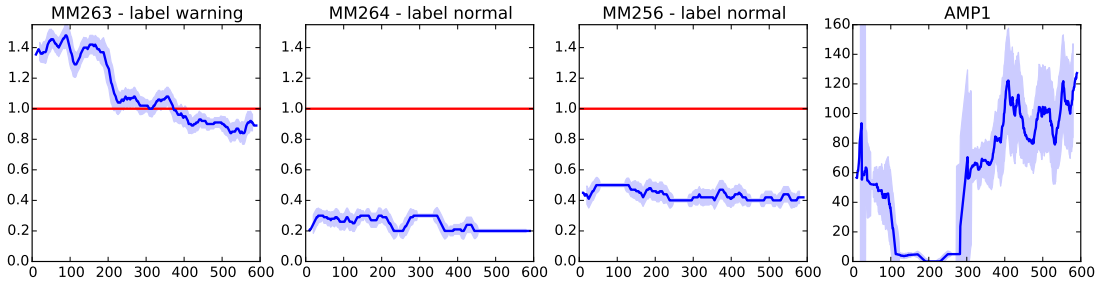


Figure A.4: Three methane detectors and ammeter located on the cutter loader. The sensor MM263 has exceeded the warning threshold.

The plots show that, apart from outliers (Figure A.1), the variation of methanometer indications is relatively small (Figure A.3). Hence, indications oscillating near the warning threshold, like in Figure A.2, are the most difficult cases. Furthermore, "warnings" were rarely indicated by more than one sensor (see example in Figure A.4) what could potentially affect on, e.g., multi-target approaches related to classifier chains technique [316].

A.2 Seismic Data

Seismic data is related to AAIA'16 Data Mining Challenge: Predicting Dangerous Seismic Events in Active Coal Mines that took place between October 5, 2015, and February 27, 2016, under auspices of 11^{th} International Symposium on Advances in Artificial Intelligence and Applications (AAIA'16) which is a part of the FedCSIS conference.

attribute no.	description
1	ID of the main working site where the measurements were taken
2	total energy of seismic bumps registered in the last 24h
3	total energy of major seismic bumps registered in the last 24h
4	total energy of destressing blasts in the last 24h
5	total seismic energy of all types of bumps
6	latest progress in the mining from, both, left and right side
7	latest seismic hazard assessment made by experts (a/b/c/d)
8	latest seismoacoustic hazard assessment by experts (a/b/c/d)
9	latest (alternative) seismoacoustic hazard assessment (a/b/c/d)
10	latest comprehensive hazard assessment made by experts (a/b/c/d)
11	maximum yield from the last meter of the small-diameter drilling
12	depth at which the maximum yield was registered
13–37	time series containing number of seismic bumps with energy in range $(0, 10^2]$ per hour (1..24)
38–61	time series containing number of seismic bumps with energy in range $(10^2, 10^3]$ per hour (1..24)
62–85	time series containing number of seismic bumps with energy in range $(10^3, 10^4]$ per hour (1..24)
86–109	time series containing number of seismic bumps with energy in range $(10^4, 10^5]$ per hour (1..24)
110–133	time series containing number of seismic bumps with energy in range $(10^5, Inf)$ per hour (1..24)
134–157	time series containing sum of energy of seismic bumps with energy in rang $(0, 10^2]$ per hour (1..24)
158–181	time series containing sum of energy of seismic bumps with energy in rang $(10^2, 10^3]$ per hour (1..24)
182–205	time series containing sum of energy of seismic bumps with energy in rang $(10^3, 10^4]$ per hour (1..24)
206–229	time series containing sum of energy of seismic bumps with energy in rang $(10^4, 10^5]$ per hour (1..24)
230–253	time series containing sum of energy of seismic bumps with energy in rang $(10^5, Inf)$ per hour (1..24)
254–277	time series containing number of seismic bumps per hour (1..24)
278–301	time series containing number of rock bursts per hour (1..24)
302–325	time series containing number of destressing blasts per hour (1..24)
326–349	time series containing energy of the strongest seismic bump per hour (1..24)
350–373	time series containing max activity of the most active geophone per hour (1..24)
374–397	time series containing max energy of the most active geophone per hour (1..24)
398–421	time series containing avg activity of the most active geophone per hour (1..24)
422–445	time series containing avg energy of the most active geophone per hour (1..24)
446–469	time series containing maximum difference in activity of the most active geophone per hour (1..24)
470–493	time series containing maximum difference in energy of the most active geophone per hour (1..24)
494–517	time series containing average difference in activity of the most active geophone per hour (1..24)
518–541	time series containing average difference in energy of the most active geophone per hour (1..24)

Table A.2: Attributes of the seismic data. The experts assessments (a/b/c/d) corresponds to: a - no hazard; b - moderate hazard; c - high hazard; d - dangerous.

All the attributes of the data set are described in Table A.2. Test and train data sets are available online at the competition's web page – AAIA'16 Data Mining Challenge at the KnowledgePit platform. To access the data it is necessary to register.

A.3 Firefighter Data

Figure A.5: Filmed and tagged training exercises synchronized with sensor readings.

The data used in the AAIA'15 data mining competition: Tagging Firefighter Activities at a Fire Scene were collected during training exercises conducted by a group of firefighters from the Main School of Fire Service in Warsaw. All cadets participating in the experiment were equipped with several sensors located on their body, including seven inertial measurement units (IMU) Polulu AltIMU-9 rev-4 with 3–axis (horizontal, vertical, and altitudinal) accelerometers with $\pm 16g$ dynamic range, and 3–axis gyroscopes with $\pm 2000 \circ /s$ maximum angular rate, and a physiological data sensor — Equivital Single Subject Kit (EQ-02-KIT-SU-4). Sensors were integrated with a data acquisition unit (DAU) on Odroid-U3+ with an external battery, additional Bluetooth, and a Wi-Fi module. The data acquisition process was further supported by XBee-PRO 868 communication nodes and Arduino micro prototype platform connected via USB to DAU.

During the exercise, cadets were simulating typical actions related to a fire incident. The video recordings of the experiment were synchronized with sensor readings and provided to domain experts, who tagged videos with the observed posture and activity. In Figure A.5, we provide a demonstrative frame from the video recording of the exercises with sample labels and selected sensor readings[1]. The data set achieved this way contains $20,000$ rows and $17,242$ columns, and is available on-line at KnowlegePit platform. Each data row corresponds to a short time series (approximately 1.8 s long) of sensor readings; hence an average time difference between consecutive sensory readings is 4.5 ms. The descriptions of data attributes and two labels are provided in Table A.3. For further reading about the measuring apparatus or the recording procedure, we may refer to the summary of AAIA'15 data mining competition [260].

[1]The video recording is available on-line on KnowlegePit platform.

attribute	description
1-42	42 attributes with aggregated statistics, including: average, standard deviation, median, skew, and avgDiff collected from physiological sensor measuring: ECG, breath rate, heart beat rate, heart rate interval (RR), and body temperature. Each attribute name ends with: ECG1, ECG2, HRBR1, HRBR2, RR, and TEMP, indicating the measured vital parameter, and each one is prefixed with the type of statistics, e.g., min_TEMP, max_ECG1, or avg_HRBR2.
time1..400	relative time of each measure (1..400)
ll_acc_x_1..400	400 readings from x-axis of accelerometers attached to each firefighter's left leg
ll_acc_y_1..400	400 readings from y-axis of accelerometers attached to each firefighter's left leg
ll_acc_z_1..400	400 readings from z-axis of accelerometers attached to each firefighter's left leg
ll_gyro_x_1..400	400 readings from x-axis of gyroscopes attached to each firefighter's left leg
ll_gyro_y_1..400	400 readings from y-axis of gyroscopes attached to each firefighter's left leg
ll_gyro_z_1..400	400 readings from z-axis of gyroscopes attached to each firefighter's left leg
rl_acc_x_1..400	400 readings from x-axis of accelerometers attached to each firefighter's right leg
rl_acc_y_1..400	400 readings from y-axis of accelerometers attached to each firefighter's right leg
rl_acc_z_1..400	400 readings from z-axis of accelerometers attached to each firefighter's right leg
rl_gyro_x_1..400	400 readings from x-axis of gyroscopes attached to each firefighter's right leg
rl_gyro_y_1..400	400 readings from y-axis of gyroscopes attached to each firefighter's right leg
rl_gyro_z_1..400	400 readings from z-axis of gyroscopes attached to each firefighter's right leg
lh_acc_x_1..400	400 readings from x-axis of accelerometers attached to each firefighter's left hand
lh_acc_y_1..400	400 readings from y-axis of accelerometers attached to each firefighter's left hand
lh_acc_z_1..400	400 readings from z-axis of accelerometers attached to each firefighter's left hand
lh_gyro_x_1..400	400 readings from x-axis of gyroscopes attached to each firefighter's left hand
lh_gyro_y_1..400	400 readings from y-axis of gyroscopes attached to each firefighter's left hand
lh_gyro_z_1..400	400 readings from z-axis of gyroscopes attached to each firefighter's left hand
rh_acc_x_1..400	400 readings from x-axis of accelerometers attached to each firefighter's right hand
rh_acc_y_1..400	400 readings from y-axis of accelerometers attached to each firefighter's right hand
rh_acc_z_1..400	400 readings from z-axis of accelerometers attached to each firefighter's right hand
rh_gyro_x_1..400	400 readings from x-axis of gyroscopes attached to each firefighter's right hand
rh_gyro_y_1..400	400 readings from y-axis of gyroscopes attached to each firefighter's right hand
rh_gyro_z_1..400	400 readings from z-axis of gyroscopes attached to each firefighter's right hand
la_acc_x_1..400	400 readings from x-axis of accelerometers attached to each firefighter's left arm
la_acc_y_1..400	400 readings from y-axis of accelerometers attached to each firefighter's left arm
la_acc_z_1..400	400 readings from z-axis of accelerometers attached to each firefighter's left arm
la_gyro_x_1..400	400 readings from x-axis of gyroscopes attached to each firefighter's left arm
la_gyro_y_1..400	400 readings from y-axis of gyroscopes attached to each firefighter's left arm
la_gyro_z_1..400	400 readings from z-axis of gyroscopes attached to each firefighter's left arm
ra_acc_x_1..400	400 readings from x-axis of accelerometers attached to each firefighter's right arm
ra_acc_y_1..400	400 readings from y-axis of accelerometers attached to each firefighter's right arm
ra_acc_z_1..400	400 readings from z-axis of accelerometers attached to each firefighter's right arm
ra_gyro_x_1..400	400 readings from x-axis of gyroscopes attached to each firefighter's right arm
ra_gyro_y_1..400	400 readings from y-axis of gyroscopes attached to each firefighter's right arm
ra_gyro_z_1..400	400 readings from z-axis of gyroscopes attached to each firefighter's right arm
torso_acc_x_1..400	400 readings from x-axis of accelerometers attached to each firefighter's torso
torso_acc_y_1..400	400 readings from y-axis of accelerometers attached to each firefighter's torso
torso_acc_z_1..400	400 readings from z-axis of accelerometers attached to each firefighter's torso
torso_gyro_x_1..400	400 readings from x-axis of gyroscopes attached to each firefighter's torso
torso_gyro_y_1..400	400 readings from y-axis of gyroscopes attached to each firefighter's torso
torso_gyro_z_1..400	400 readings from z-axis of gyroscopes attached to each firefighter's torso
posture	a label describing a posture of a firefighter.
activity	a label describing a main activity of a firefighter.

Table A.3: Attributes of the firefighter data.

A.4 AWS Spot Data

The data were iteratively collected over the period between November 11, 2019 and March 11, 2020, from 11 AWS regions[2]: *ap-northeast-1* (427309), *ap-northeast-2* (303113), *ap-south-1* (322593), *ap-southeast-1* (459592), *ap-southeast-2* (382703), *ca-central-1* (206218), *eu-central-1* (453457), *eu-west-1* (546474), *sa-east-1* (280076), *us-east-1* (1061429), *us-west-1* (285975), and *us-west-2* (661369) using AWS command line interface (CLI v2) with the following command:

```
aws2 ec2 describe-spot-price-history
--region <e.g., us-east-1>
--start-time  <e.g., 2019-11-11T12:00:00>
--end-time <e.g., 2019-11-18T12:00:00>
--output text
```

The raw data consisted of a total of $5.4M$ unique records, each corresponding to a bid for one of the AWS spot instances, in a form as presented in Table A.4. The preliminary data exploration revealed that the spot price time series for a given instance type differ between regions and availability zones.

Table A.4: Exemplary spot price bids collected from AWS.

SPOTPRICEHISTORY	Region & AZ	Instance Type	System	Bid Price	Bid Date & Time
SPOTPRICEHISTORY	sa-east-1c	m4.xlarge	Linux/UNIX	0.076300	2020-02-11T14:50:42+00:00
SPOTPRICEHISTORY	sa-east-1b	m5.large	Linux/UNIX	0.052800	2020-02-11T14:25:45+00:00
SPOTPRICEHISTORY	sa-east-1c	m5.24xlarge	Linux/UNIX	2.055900	2020-02-11T14:11:04+00:00
SPOTPRICEHISTORY	sa-east-1c	m5.24xlarge	Windows	6.471900	2020-02-11T14:10:52+00:00
SPOTPRICEHISTORY	sa-east-1a	r3.2xlarge	SUSE Linux	0.262800	2020-02-11T14:08:50+00:00

After the initial filtering and pre-processing, the data were aligned into 854 time series. One per each *region*, *AZ*, *instance type* triple, and aggregated daily as presented in Table A.5. *Volume* column presents the number of price changing bids recorded in a given time window. *Open* refers to the first bid in a given time window. Whereas *High*, *Low*, and *Close* columns refer to highest, lowest, and the last bid in each time window, respectively. If no bids were recorded, i.e., *Volume* equals to zero (cf. last row in Table A.5), *Open*, *High*, *Low*, and *Close* rates were assigned the same value as the *Close* price in the previous window. Such a data format allowed us to represent time series as candlestick charts (cf. Figure 5.14).

Table A.5: Data aggregated in 24h long time windows starting at *Window Begin*.

Region	AZ	Instance Type	Window Begin	Open ($)	High ($)	Low ($)	Close ($)	Volume
us-east-1	d	r5.12xlarge	2020-03-07 12:00:00	0.8790	0.8790	0.8763	0.8763	3
us-east-1	d	r5.12xlarge	2020-03-08 12:00:00	0.8761	0.8761	0.8757	0.8757	4
us-east-1	d	r5.12xlarge	2020-03-09 12:00:00	0.8767	0.8810	0.8767	0.8810	3
us-east-1	d	r5.12xlarge	2020-03-10 12:00:00	0.8799	0.8882	0.8799	0.8863	4
us-east-1	d	r5.12xlarge	2020-03-11 12:00:00	0.8863	0.8863	0.8863	0.8863	0

[2]Numbers in brackets indicate the amount of unique bids in data for each region.

Appendix B

B.1 Expert methods for classifications of seismic hazards in coal mines

Two basic methods are routinely used by experts for the assessment of seismic hazards in Polish coal mines. These methods are often called *seismic* and *seismoacoustic*, respectively [197].

The essence of the seismic method is the analysis of tremor occurrences in mines. Table B.1 presents the basis for quantitative hazard assessment using this method. This type of assessment is performed routinely every shift. As shown in Table B.1, very simple and intuitive rules are used in order to model the relationship between the energy of tremors and rock bursts. These rules were designed by experts based on their experience and common sense.

The seismoacoustic method is based on an analysis of seismoacoustic emissions recorded at a given longwall. The seismoacoustic emission is described by its intensity, understood as the number of registered events and their total energy. The dependency between the seismoacoustic emission seismic hazards was often observed in practice by mining experts. In this type of assessment, the following factors are considered as crucial:

- registered seismoacoustic emissions,
- the number of pulses recorded by geophones, which is converted into so-called conventional seismic energy using an appropriate formula.

Available studies on the effectiveness of the seismic and seismoacoustic methods are limited to those conducted by the Polish Central Mining Institute [196]. Its analysis of selected cases of rock-bursts showed that the seismic method correctly predicted these dangerous events in only about 17% of cases. When the seismic method was coupled with the seismoacoustic approach (i.e., a hazardous state is predicted when any of the methods indicate the state '**d**'), the prediction accuracy increased to about 20%. However, the data set used for the purpose of that experiment was relatively small and did not cover coal mines located in different geographical locations. The used data set is not publicly available, hence the results of this evaluation were difficult to reproduce.

Table B.1: Quantitative assessments of seismic hazards based on the observed seismic activity, as outlined in [197].

Rockburst hazard	Caved faces	Roadways
a no hazard	1. No tremors or single tremors with energies E of the order of 10^2J – 10^3J 2. $E_{max} \leq 10^4$J 3. $\Sigma E < 10^5$J per 5m of longwall advance	1. No tremors or single tremors with energies E of the order of 10^2J 2. $E_{max} \leq 10^3$J 3. $\Sigma E < 10^3$J per 5m of longwall advance
b low hazard	1. Occurrence of tremors with energies E of the order of 10^2J – 10^5J 2. 10^4J $< E_{max} \leq 10^5$J 3. $10^5 \leq \Sigma E < 10^6$J per 5m of longwall advance	1. Occurrence of single tremors or single tremors with energies E of the order of $10^2 - 10^3$J 2. $E_{max} \leq 5 \cdot 10^3$J 3. 10^3J$\leq \Sigma E < 10^4$J per 5m of longwall advance
c moderate hazard	1. Occurrence of tremors with energies E of the order of 10^2J – 10^6J 2. $5 \cdot 10^5$J $< E_{max} \leq 5 \cdot 10^6$J 3. $10^6 \leq \Sigma E < 10^7$J per 5m of longwall advance	1. Occurrence of single tremors or single tremors with energies E of the order of $10^2 - 10^4$J 2. $5 \cdot 10^3$J $\leq E_{max} \leq 5 \cdot 10^5$J 3. 10^4J $\leq \Sigma E < 10^5$J per 5m of longwall advance
d high hazard	1. Occurrence of tremors with energies E of the order of 10^2J – 10^6J 2. $E_{max} > 5 \cdot 10^6$J 3. $\Sigma E > 10^7$J per 5m of longwall advance	1. Occurrence of single tremors or single tremors with energies E of the order of $10^2 - 10^5$J 2. $E_{max} > 10^5$J 3. $\Sigma E > 10^5$J per 5m of longwall advance