Order in the random forest

Isak Karlsson

Academic dissertation for the Degree of Doctor of Philosophy in Computer and Systems Sciences at Stockholm University to be publicly defended on Thursday 8 June 2017 at 13.00 in L30, NOD-huset, Borgarfjordsgatan 12.

Abstract

In many domains, repeated measurements are systematically collected to obtain the characteristics of objects or situations that evolve over time or other logical orderings. Although the classification of such data series shares many similarities with traditional multidimensional classification, inducing accurate machine learning models using traditional algorithms are typically infeasible since the order of the values must be considered.

In this thesis, the challenges related to inducing predictive models from data series using a class of algorithms known as random forests are studied for the purpose of efficiently and effectively classifying (i) univariate, (ii) multivariate and (iii) heterogeneous data series either directly in their sequential form or indirectly as transformed to sparse and high-dimensional representations. In the thesis, methods are developed to address the challenges of (a) handling sparse and high-dimensional data, (b) data series classification and (c) early time series classification using random forests. The proposed algorithms are empirically evaluated in large-scale experiments and practically evaluated in the context of detecting adverse drug events.

In the first part of the thesis, it is demonstrated that minor modifications to the random forest algorithm and the use of a random projection technique can improve the effectiveness of random forests when faced with discrete data series projected to sparse and high-dimensional representations. In the second part of the thesis, an algorithm for inducing random forests directly from univariate, multivariate and heterogeneous data series using phase-independent patterns is introduced and shown to be highly effective in terms of both computational and predictive performance. Then, leveraging the notion of phase-independent patterns, the random forest is extended to allow for early classification of time series and is shown to perform favorably when compared to alternatives. The conclusions of the thesis not only reaffirm the empirical effectiveness of random forests for traditional multidimensional data but also indicate that the random forest framework can, with success, be extended to sequential data representations.

Keywords: *Machine learning, random forest, ensemble, time series, data series, sequential data, sparse data, high-dimensional data.*

Stockholm 2017
http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-142052

ISBN 978-91-7649-827-9
ISBN 978-91-7649-828-6
ISSN 1101-8526

Stockholm University

Department of Computer and Systems Sciences

Stockholm University, 164 07 Kista

Order in the random forest

Isak Karlsson

Order in the random forest

Isak Karlsson

ISBN (printed): 978-91-7649-827-9
ISBN (digital): 978-91-7649-828-6
DSV Report Series No. 17-004
ISSN 1101-8526
Printed in Sweden by Universitetsservice US-AB, Stockholm 2017

Distributor: Department of Computer and Systems Sciences, Stockholm University

Abstract

In many domains, repeated measurements are systematically collected to obtain the characteristics of objects or situations that evolve over time or other logical orderings. Although the classification of such data series shares many similarities with traditional multidimensional classification, inducing accurate machine learning models using traditional algorithms are typically infeasible since the order of the values must be considered.

In this thesis, the challenges related to inducing predictive models from data series using a class of algorithms known as random forests are studied for the purpose of efficiently and effectively classifying (i) univariate, (ii) multivariate and (iii) heterogeneous data series either directly in their sequential form or indirectly as transformed to sparse and high-dimensional representations. In the thesis, methods are developed to address the challenges of (a) handling sparse and high-dimensional data, (b) data series classification and (c) early time series classification using random forests. The proposed algorithms are empirically evaluated in large-scale experiments and practically evaluated in the context of detecting adverse drug events.

In the first part of the thesis, it is demonstrated that minor modifications to the random forest algorithm and the use of a random projection technique can improve the effectiveness of random forests when faced with discrete data series projected to sparse and high-dimensional representations. In the second part of the thesis, an algorithm for inducing random forests directly from univariate, multivariate and heterogeneous data series using phase-independent patterns is introduced and shown to be highly effective in terms of both computational and predictive performance. Then, leveraging the notion of phase-independent patterns, the random forest is extended to allow for early classification of time series and is shown to perform favorably when compared to alternatives. The conclusions of the thesis not only reaffirm the empirical effectiveness of random forests for traditional multidimensional data but also indicate that the random forest framework can, with success, be extended to sequential data representations.

Sammanfattning

Inom många områden genereras kontinuerligt stora mängder mätdata som innehåller information om olika företeelser och hur dessa förändras över tid (eller över någon annan logisk ordning). Även om metoder för klassificeringen av sådana dataserier delar många likheter med traditionella metoder för klassificering av flerdimensionella data, så är traditionella algoritmer ofta ineffektiva eftersom de inte tar mätvärdenas ordning i beaktande.

I den här avhandlingen studeras utmaningar relaterade till effektiv konstruktion av random forest-modeller från dataserier för att klassificera (i) univariata, (ii) multivariata och (iii) heterogena dataserier antingen direkt i sin sekventiella form eller indirekt som transformerade till glesa och högdimensionella representationer. I avhandlingen utvecklas metoder för att med random forests (a) hantera glesa och högdimensionella data, (b) hantera klassificering av dataserier och (c) hantera tidig klassificering av tidsserier. De föreslagna metoderna utvärderas empiriskt i flera experiment och dess praktiska värde utvärderas för detektion av läkemedelsbiverkningar i patientjournaldata.

I avhandlingens första del visas hur mindre modifikationer av random forest-algoritmen samt hur användandet av en projektionsteknik kan förbättra den prediktiva prestandan hos random forest-modeller, när dessa konstrueras från diskreta dataserier som projicerats till högdimensionella och glesa representationer. I avhandlingens andra del introduceras en algoritm för att konstruera random forest-modeller direkt från univariata, multivariata och heterogena dataserier genom att använda fasoberoende och slumpmässigt utvalda mönster. Den algoritmer som presenteras visar sig, i flera utvärderingar och jämförelser, vara mycket effektiva vad gäller både beräkningsmässig och prediktiv prestanda. Vidare, genom att nyttja fasoberoende mönster, utökas slutligen random forest-algoritmen för att generera modeller som möjliggör tidig klassificering av tidsserier. För tidig klassificering av tidserier visar en empirisk utvärdering att den introducerade algoritmen presterar väl i jämförelse med alternativa metoder. Avhandlingens huvudsakliga slutsats bekräftar dels effektiviteten hos random forest-modeller för traditionella flerdimensionella data, men de empiriska utvärderingarna visar också att random forest-ramverket med framgång kan utvidgas till sekventiella datapresentationer.

List of Papers

This thesis summarizes several contributions. The following publications, referred to in the text by their Roman numerals, are discussed in this thesis.

PAPER I: Karlsson, I., Zhao, J., Asker, L., Boström, H. (2013) *Predicting Adverse Drug Events By Analyzing Electronic Patient Records*, In: Proceedings of Conference on Artificial Intelligence in Medicine, pp. 125–129. doi:10.1007/978-3-642-38326-7_19

PAPER II: Karlsson, I., Zhao, J. (2014) *Dimensionality Reduction with Random Indexing: an application on adverse drug event detection using electronic health records*, In: Proceedings of Computer-Based Medical Systems, pp. 304–307. © 2014 IEEE. doi:10.1109/CBMS.2014.22

PAPER III: Karlsson, I., Bostrom, H. (2014) *Handling Sparsity with Random Forests when Predicting Adverse Drug Events from Electronic Health Records*, In: Proceedings of International Conference on Healthcare Informatics, pp. 17–22. © 2014 IEEE. doi:10.1109/ICHI.2014.10

PAPER IV: Karlsson, I., Bostrom, H., Papapetrou, P. (2015) *Forests of Randomized Shapelet Trees* In: Proceedings of International Symposium on Learning and Data Sciences, pp. 126–136. doi:10.1007/978-3-319-17091-6_8

PAPER V: Karlsson, I., Papapetrou, P., Asker, L. (2015) *Multi-channel ECG classification using forests of randomized shapelet trees*, In: Proceedings of International Conference on Pervasive Technologies Related to Assistive Environments, pp. 258–262. doi:10.1145/2769493.2769520

PAPER VI: Karlsson, I., Papapetrou, P., Boström, H. (2016) *Generalized Random Shapelet Forest*, Data Mining and Knowledge Discovery, Vol. 30, No. 5, pp. 1053–1085. doi:10.1007/s10618-016-0473-y

PAPER VII: Karlsson, I. Boström, H. (2016) *Predicting Adverse Drug Events using Heterogeneous Event Sequences*, In: Proceedings of International Conference on Healthcare Informatics, pp. 356–362. © 2016 IEEE. doi:10.1109/ICHI.2016.64

PAPER VIII: Karlsson I., Papapetrou, P., Boström, H., (2016) *Early Random Shapelet Forest*, In: Proceedings of Discovery Science, pp. 261–276 (*Carl H. Smith Award* for best paper). doi:10.1007/978-3-319-46307-0_17

Reprints were made with permission from the publishers.

Related Papers

During the course of this thesis, several fruitful collaborations have resulted in the following papers, not discussed in the thesis.

PAPER IX: Kotsifakos A., Karlsson I., Papapetrou P., Athitsos V., and Gunopulos D. (2015) *Embedding-based Subsequence Matching with Gaps-Range-Tolerances: a Query-By-Humming application*, Very Large Databases Journal, Vol. 30, No. 5, pp. 519–536

PAPER X: Henelius A., Karlsson I., Papapetrou P., Ukkonen A., Puolamäki K. (2016) *Semigeometric Tiling of Event Sequences*, In: Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, pp. 329–344

Author's contribution

In the publications included in this thesis where I am the leading author, the main contribution of the study is mine. In these cases, I have led the research, including the development of the algorithms, the design of the experiments, and authoring the main parts of the manuscript. In Paper **I**, and Paper **II** the two lead authors made equal contributions regarding experiment design and manuscript editing. In all included publications, the authors have discussed and approved the final manuscript.

Acknowledgements

I would chiefly like to thank my adviser Henrik Boström, co-adviser Lars Asker, and collaborator Panagiotis Papapetrou, who made my time as a graduate student an immensely meaningful experience. You taught me research. Thanks also to the many colleagues that have provided valuable contributions to this work. In particular I would like to thank Jing Zhao, Aron Henriksson, Karl Jansson, Ram Gurung, and the rest of the data science group for fruitful discussions and research collaborations. Also thanks to Tobias, Beatrice, Samuel, Irvin and the rest of the department for making this the best possible journey. Last (but not least) I would like to extend my deepest love and gratitude to Lisa, without whose constant support, encouragement and discussions this dissertation would never have existed.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

Due to the increased digitalization and improved computing power, ever-expanding amounts of data, from social media interactions and advertisement patterns to industrial processes and medical records, are generated globally every day. At hospitals, for instance, the entire medical history of the patients, including what drugs have been prescribed, what diagnoses have been assigned, and what laboratory tests have been conducted, are continually recorded in electronic health record (EHR) systems. As these data sources grow increasingly more complex, with each object being represented by thousands, or perhaps even millions of variables, each of which can change over time, finding relevant and accurate patterns that can be used for predictive models is impractical for human experts. Alongside this "data revolution" [Kitchin, 2014; Mayer-Schönberger and Cukier, 2013], increased capabilities of computing hardware and theoretical developments in computer science, artificial intelligence, and statistics, have made computers able to *learn from data* and uncover the predictive structure of problems. In particular, such techniques have arisen from the field of *machine learning* [Abu-Mostafa et al., 2012; Mitchell, 1997].

Machine learning is defined as the study of algorithms that can learn a generalizable model from the data for a problem without being explicitly programmed to solve that particular problem. More formally, a machine learning algorithm is said to learn from past experiences (data) with respect to some class of tasks and performance metric, if the performance of those tasks improves with experience [Mitchell, 1997]. To give an example, suppose that an airline receives thousands of ticket refund requests every day, and one wants to automate the process of evaluating them. Further, suppose that there is no formula for when a request is eligible for a refund, but there is an abundance of data from previous cases where refunds have been either rejected or approved. In this example, the database of past refund cases is known as the *training set* and each case is an *instance* (this term is used interchangeably with *object* or *example*). Furthermore, each refund request is represented in a suitable way, e.g., using personal information related to the flight in question, such as membership status, flight information, payment method, etc., which are known as

the *attributes* of the multidimensional data. Each record also keeps track of the refund status of each case: either approved or rejected, which in this case can be described as the *target variable*. The idea is then that the chosen learning algorithm, guided by the training set, fits a successful *predictive model*. This model is subsequently used to predict the label value of the target variable (e.g., accept or reject) of a previously unseen example (e.g., a refund request). The primary goal when constructing predictive machine learning models, as captured by its definition, is that the induced models perform well under some specified performance metric: not only for the data that was used during training, but also for new data that has not previously been seen by the algorithm.

Similar to the flight refund example presented earlier, most standard machine learning algorithms require the data to be represented in a multidimensional (tabular) format, where each object is represented as a vector of static attributes. However, unlike multidimensional data, in which all attributes are treated equally, many recent data sources contain variables that are logically ordered, for instance, evolving over time. Such data are generally referred to as *data series*[1] (when the logical ordering refers to time, the term time series is used interchangeably). For example, the flight ticket refund case might contain variables with temporal dependencies, such as, the history of the flight, payments, etc., which affect the choice of learning algorithm and impact the resulting predictive models.

In most cases, the data series consists of a single numerical measurement (referred to as a dimension), e.g., heart rate, but in many applications, the objects are composed of multiple measurements, making these data series *multivariate* (d-dimensional) as opposed to *univariate* (1-dimensional). For multivariate data series, the individual dimensions are either of the same type, e.g., real-valued or discrete, or of different types. Such data series are referred to as *homogeneous* and *heterogeneous*, respectively. Although the values of data series are more interesting than their ordering from a domain perspective, the values cannot be properly interpreted without information about their order. For example, consider measurements of blood pressure. For data series of such measurements, the value of the tests might be considered to be more interesting than when the tests were performed, but information about changes in the measurements, such as increases or decreases, cannot be properly understood without information about the order. Moreover, since blood pressure is an example of a multivariate data series consisting of measures of both the diastolic and the systolic pressure, its proper assessment requires the analysis of the structures both within a single dimension and between several dimensions.

Data series classification shares many similarities with traditional multidi-

[1]Recently, an alternative term for data series is *contextual representation* [Aggarwal, 2015].

mensional classification. However, inducing accurate machine learning models from data series data using traditional algorithms is in many cases unfeasible [see Bagnall et al., 2016]. Specifically, the predictive models induced from data series are, unlike models that are learned from multidimensional data, dependent not only on a combination of interrelations between the values within a single dimension but also between several dimensions. To account for the impact of sequentially ordered data, machine learning algorithms must typically be significantly altered [Aggarwal, 2015; Xing et al., 2010].

1.2 Data series classification

To construct traditional predictive machine learning models from data series, several common representations can be employed. One common representation, especially for discrete data series, is to use binary vector representations in which events are extracted from the data series in order to create a multidimensional vector space where the presence or absence of items is recorded [Kudenko, 1998; Lesh et al., 1999]. For many such representations, however, the number of discrete events is large, and the number of attributes in the resulting datasets increases with the complexity of the transformation procedure, which results in high-dimensional vectors where most attributes take on the value zero (i.e., absent) and few attributes have non-zero values (i.e., present). This phenomenon is commonly known as *data sparsity* and has been shown to significantly impact the performance of most standard machine learning algorithms [Allison et al., 2006; Goldstein et al., 2010].

Although several methods to handle high-dimensional and sparse data have been considered, e.g., by filtering and selecting attributes, there are often a few objects with a large number of attributes available during training, making it statistically intractable to identify the most important ones [Yu and Liu, 2003]. Therefore, other approaches have tried to remove and compress redundant attributes using dimensionality reduction, where the original (high) dimensional space is projected to a new lower-dimensional space. The design and utilization of such systems, however, present many challenges. Specifically, they often scale poorly with an increasing number of attributes and instances [Saeys et al., 2007] and require multiple choices to be evaluated and compared. Another approach to address the problem of data sparsity in high dimensions is to handle the high-dimensionality directly in the learning algorithm [Caruana et al., 2008; Shevade and Keerthi, 2003], thereby reducing the number of hyper-parameters and pre-processing methods that require optimization.

An alternative, and complementary method to deal with data series classification is to utilize machine learning algorithms that have been developed directly to handle global and local similarities in such sequences. To this end,

several machine learning methods have been introduced and evaluated from various perspectives, e.g., prototype-based methods [Xing et al., 2010], which use various distance measures to perform predictions. Currently, the vast majority of the state-of-the-art methods for data series classification focus on univariate data series [Xing et al., 2010]. Concurrently, however, multivariate, and even heterogeneous, data series are becoming increasingly important in many domains [e.g., Maharaj and Alonso, 2014].

Prototype-based methods predominately focus on classification problems where class membership is determined by a common pattern along the order axis, and the variations are caused by noise in the observations or shifts along the order axis. However, in many cases there are smaller shapes which can begin at any point along the order axis, i.e., phase independent shapes that define class memberships. In cases where class membership is determined by global phase-independent similarities, frequency domain features can be used to induce effective predictive models [Esling and Agon, 2012]. On the other hand, if the discriminatory subsequence is local, then frequency domain features are insufficient to define class membership [see Ye and Keogh, 2009].

To address the problem of local discriminatory shapes, recent approaches for the classification of numerical data series rely on the identification and extraction of subsequences, referred to as *shapelets* [Ye and Keogh, 2009]. In the literature, several shapelet-based approaches have been proposed, most notably shapelet-based decision trees [Ye and Keogh, 2011] and feature extraction approaches using shapelet transformations [Hills et al., 2014]. Similar approaches have also been proposed for multivariate data series [see Cetin et al., 2015; Patri et al., 2014]. Unfortunately, however, it has been demonstrated that while shapelet-based decision trees can provide interpretable rules and, hence, insights to practitioners, their classification accuracy, and training costs are often prohibitive. In particular, the computational cost often limits their applicability when dealing with large and multivariate data series. Moreover, current methods for multivariate data series do not take into account correlations between different dimensions, something which has been shown to have significant importance [see Bankó and Abonyi, 2012]. Hence, although there are several methods for learning predictive models from both univariate and multivariate data series, these predictive models cannot effectively handle large databases of univariate, multivariate and heterogeneous data series in an efficient and effective manner.

In addition to the complexity of constructing efficient and effective predictive models from data series, recent advances have led to the investigation of methods that can provide predictions in streaming settings, where the data of the objects is received in temporal order [Xing et al., 2008]. Thus, since the values of a particular object are continuously received, e.g., in time-stamp

ascending order, it is important and indeed desirable to monitor and classify the time series as early as possible. In cases such as these, there is a trade-off between *earliness*, i.e., how early the prediction can be made, and *accuracy*, i.e., the rate of correct assessments. To address these types of classification problems, a plethora of algorithms have been proposed for the early prediction of time series [e.g., Dachraoui et al., 2015; Mori et al., 2016; Xing et al., 2008]. Although these methods can provide predictions early, they require the induction of several predictive models (one for each time-step), which reduces the inductive efficiency and, hence, applicability to large databases of time series.

Depending on the task, the performance metric, and the data at hand, there are several possible choices of learning algorithms, each of which has numerous hyper-parameters, induction algorithms, and inherent learning biases, which result in different performance characteristics. This is usually expressed in terms of the the "no free lunch theorem" [Wolpert, 1996], which states that for any given domain and application there is no single machine learning algorithm that always induces the most accurate predictive model. The usual approach is, thus, to try a number of algorithms and choose the one that performs best on a separate validation set. Although the performance of machine learning algorithms can usually be fine-tuned to improve the predictive performance of a given task. This optimization is often computationally intractable, complex, and error-prone. Instead, by combining the predictions of several models, with the idea that the mistakes of one model can be complemented by another, the predictive performance has, both theoretically and empirically, been shown to improve [Mitchell, 1997; Opitz and Maclin, 1999].

Models that combine the output of several predictive (base-)models are known as *ensemble* methods. Random forests (originally introduced by Breiman [2001]) constitute an extremely successful, and beautifully simple family of predictive models that constructs a *forest* (an ensemble) of randomized decision trees that introduces randomness in both the training data and in the base models. Due to its often state-of-the-art predictive performance, simplicity, and reliability, random forests, in various forms, have been employed for a wide variety of tasks [Fernández-Delgado et al., 2014]. However, random forests have, traditionally, been constrained to multidimensional data, limiting their applicability when considering large databases of possibly heterogeneous and multivariate data series.

1.3 Research problem

In summary, the main problem studied in this dissertation concerns the design and analysis of a class of algorithms known as random forests for the purpose of efficiently (i.e., with low computational cost) and effectively (i.e., with high predictive performance) classify (i) univariate, (ii) multivariate and (iii) heterogeneous data series either directly in their sequential form or indirectly when transformed to high-dimensional and sparse representations. In doing so, certain design decisions need to be considered and empirically evaluated. More specifically, although random forests have been shown to be a robust, accurate and successful tool for numerous prediction tasks [see Fernández-Delgado et al., 2014], there are still many unknown areas regarding their applicability to data representations that suffer from high sparsity and dimensionality, or contain temporal or other similarly ordered dependencies. This thesis aims to bridge this gap by solving some of the technical challenges involved in constructing random forests for data series classification, including:

a) *High-dimensional and sparse.* While random forests are known to handle high-dimensional data, the resulting predictive models are susceptible to majority-class bias in cases when said data is highly sparse, due to various implementation details [see Amaratunga et al., 2008]. By adjusting for this bias the resulting predictive models provide predictions that more accurately reflect the prior class distribution. However, it remains unclear how to design random forests in a way that considers the trade-off between efficiency and effectiveness when handling high-dimensional and sparse data which can, e.g., be the result when transforming discrete data series to multidimensional representations.

b) *Univariate, multivariate and heterogeneous data series.* Although a multitude of algorithms have been introduced for univariate data series classification [see Bagnall et al., 2016; Xing et al., 2010], many of these approaches either do not take localized and phase-independent patterns into consideration or while doing so require significant computational resources [e.g., Grabocka et al., 2014; Hills et al., 2014; Ye and Keogh, 2009]. Similarly, multivariate data series are characterized not only by (local) similarities within single dimensions but by relations between different dimensions of varying importance. The latter is not captured by traditional univariate or multivariate approaches [see Bankó and Abonyi, 2012]. Moreover, while multivariate data series can be heterogeneous with dimensions of variable types, limited attention has been given to the classification of such data series. In summary, it has remained unclear how to incorporate local, phase-independent, patterns to construct random forests for (i) univariate,

(ii) multivariate and (iii) heterogeneous data series classification efficiently and effectively.

c) *Early prediction.* In early prediction it is natural to assume that the earliness of the predictions is related to the accuracy of the predictions, since assessing more data points stabilizes the predictions and because more complete information about the time series becomes available. In this light, early prediction of data series concerns the identification of the trade-off between two conflicting properties: the earliness of the predictions and their accuracy. Due to the localization of phase-independent patterns, models that rely on such features naturally extend to early prediction. However, it has remained unclear how to design random forests that use such patterns to induce predictive models which can efficiently and effectively take into consideration the trade-off between earliness and predictive performance.

The practical motivation for this thesis originates in the medical domain, which has over the last decade seen a tremendous increase and interest in methods for making inferences about patient care using large quantities of medical data, often stored in electronic health records. As such, the electronic health records contain complex data series of the patient's medical history which capture all kinds of data related to patient care, such as medications, diagnoses and laboratory tests. Despite its great potential and interest, the predictive analysis of EHR data has not yet seen widespread adoption. One reason for this is the existence of many challenges introduced when analyzing such data using traditional machine learning algorithms [see Jensen et al., 2012]. For example, medical data has many attributes (high-dimensional), but few measurements (sparse), and these attributes are heterogeneous, encompassing categorical, quantitative, and ordered data that contain temporal dependencies. In essence, EHR data represents a prime example of complex data series for which there are many important applications of predictive modeling.

1.4 Research question

Based on the problem outlined earlier, the research question this dissertation addresses is thus formulated as: **how can random forests be created to support the efficient and effective classification of data series?**

Inducing efficient and effective random forests for data series classification can be studied from several different perspectives. Since the studies in this thesis focus on some aspects, the main research question is answered by addressing the problems outlined earlier. Consequently, in an attempt to address the overall research question, the following objectives have been set:

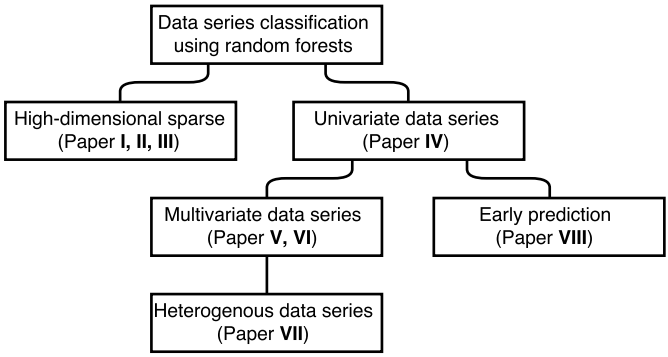

Figure 1.1: Overview of the included studies and their relation to the research problem.

a) Investigate how random forests can be extended to support efficient and effective classification of high-dimensional and sparse data representations, which, e.g., result from transforming discrete data series to a multidimensional format.

b) Investigate how the random forest algorithm can be extended to support (i) univariate, (ii) multivariate and (iii) heterogeneous data series classification efficiently and effectively.

c) Investigate how the random forest can be extended to support the early prediction of univariate data series in a way that provides an effective trade-off between earliness and predictive performance efficiently.

In this thesis, the effectiveness and efficiency of a predictive model are used as comparative statements. This means that the absolute performance (as defined in Section 3) of the predictive models is not evaluated, but rather the relative performance of several alternatives. The exact definition of efficiency and effectiveness depends on the task and choice of performance measure (see Section 3.2.1).

1.5 Contributions

The main contributions, published in eight studies, of this thesis relate to the design and evaluation of a class of machine learning algorithms known as random forests for the purpose of (early) data series classification which can be either univariate, multivariate or heterogeneous. The presented algorithms are domain independent and evaluated as such, but to affirm their utility some of

the algorithms have also been applied to the task of detecting adverse drug events in medical health records.

Figure 1.1 shows an illustrative representation of how the included studies are connected to providing an answer to the research question. Below, a summary of the contents and contributions of each paper is given.

PAPER I: **Predicting adverse drug events by analyzing electronic patient records**

This paper studies the feasibility of predicting adverse drug events from structured data extracted from electronic health records using random forests. The proposed method, to represent medical history data as sparse binary vectors, is shown to give relevant predictions, evaluated both using model effectiveness and medical expertise. The paper highlights several challenges related to the research question of this dissertation and acts as motivation for the outlined objectives and sub-questions. The first challenge motivates Paper **II** and **III** and concerns the use of random forests to efficiently and accurately handle medical data series projected to high-dimensional and sparse representations. The second challenge motivates **VII** and concerns how to effectively and efficiently induce a random forest from heterogeneous data series. Finally, the third challenge motivates Paper **VIII** and involves strategies for inducing predictive models from data series that are also able to provide early predictions.

PAPER II: **Dimensionality reduction with random indexing: an application on adverse drug event detection using electronic health records**

This paper studies the problem of high-dimensional representations introduced by transforming medical data series to a multi-dimensional format. That is, instead of handling the introduced problem of sparsity directly in the learning algorithm (as in Paper **III**), the discrete events are projected, using a method adapted from the text mining community, to a lower dimensional space, effectively ignoring the temporal dimension, which is then utilized by the random forest algorithm to induce predictive models. The study contributes towards the first objective by presenting an experimental investigation of the proposed strategy. The empirical investigation suggests that by reducing the dimensionality using the presented approach, the predictive performance and

computational efficiency, when predicting adverse drug events (ADE), can be significantly improved.

PAPER III: **Handling sparsity with random forest when predicting adverse drug events from electronic health records**

Instead of handling the sparse and high-dimensional representations separates from the model construction, and thereby introduce additional hyper-parameters that require optimization, this study introduces and evaluates several approaches for handling the high-dimensional and sparse data directly in the classification trees of the random forests. Thus, this study contributes towards the first objective. Experiments conducted on an extensive collection of datasets extracted from an electronic health record system show that the introduced strategies significantly increase the predictive performance compared to random forests without the proposed modifications. Furthermore, a bias–variance decomposition of the conditional probabilities is investigated to explain the differences in effectiveness among the considered strategies.

PAPER IV: **Forests of randomized shapelet trees**

One problem, introduced in Paper **I**, related to predicting adverse drug events from medical records, is the temporal nature of a patient history, e.g., the evolution of drug concentrations in the blood. In this study, a random forest algorithm for handling univariate numerical data series is introduced and evaluated on a number of standard benchmark datasets and, thus, contributes towards the second objective. The proposed algorithm builds an ensemble of shapelet-based decision trees using a random selection of sub-sequences. An empirical investigation shows that the introduced algorithm significantly outperforms related methods in terms of classification accuracy. The study also, although not directly related to the research question of the thesis, presents an approach for interpreting the resulting forests.

PAPER V: **Multi-channel ECG classification using forests of randomized shapelet trees**

In many medical domains, e.g., when predicting adverse drug events or other medical conditions as motivated by Paper **I**, the object to be classified does not only consist of univariate data series (as expected by the algorithm presented in Paper **IV**). Rather, such data often consists of multivariate data series. This study

presents and investigates a split-and-combine framework for classifying such data series using an ensemble approach. The ensemble approach consists of building a single model per dimension and combining the resulting models using various standard techniques. The empirical results show that combining several variables in such a way indeed improves the predictive performance over using any single variable, and that using the algorithm presented in Paper **IV** as the base-classifier outperforms the use of a nearest neighbor base-classifier. This contribution goes towards the second objective.

PAPER VI: **Generalized random shapelet forest**

In this paper, a generalization of the decision forest introduced in Paper **IV** is extended to handle objects consisting of multivariate data series. The algorithm presented in the study involves sampling both the dimension and the sub-sequence at random, constructing shapelet-based decision trees that can consider relations both within and across dimensions attributes. This strategy is orthogonal to the split-and-combine approach presented in Paper **V**, since the number of ensemble members is not limited by the data, and since it can prioritize important dimensions. The resulting algorithm is extensively evaluated, and compared to a broad range of both univariate and multivariate predictive models over a large selection of benchmark datasets, to address and provide an answer to the third sub-question. The results indicate that the presented approach results in one of the strongest predictive models available for multivariate data series. The empirical performance of the algorithm is explained using the bias–variance trade-off.

PAPER VII: **Predicting adverse drug events using heterogeneous events sequences**

In this paper, the generalized random shapelet forest presented in Paper **VI** is further extended to support heterogeneous data series consisting of both discrete and numerical values. The extension is evaluated in a setting similar to that of Paper **I**, that is, to predict the presence or absence of ADEs. To investigate the feasibility of the proposed strategy, it is compared to the best performing approach presented in Paper **III**. An experimental investigation indicates that the suggested temporal pattern forest, which considers the temporal nature of the electronic health record data,

outperforms random forests that do not. This paper thus contributes towards the second objective.

PAPER VIII: **Early random shapelet forest**

In this study, an extension of the random forest presented in Paper **IV** and **VI** is introduced in direct response to the third objective, i.e., to support the early prediction of data series. The study presents a random forest algorithm that allows tuning the trade-off between accuracy and earliness. The algorithm thereby supports the generation of early random forests that can be dynamically and efficiently adapted to the specific needs of various applications, at low computational cost. An empirical investigation indicates that the proposed strategy is more effective than alternative approaches, as measured by predictive performance, while at the same time utilizing less computational resources.

1.6 Disposition

In Chapter 2, a comprehensive review of supervised learning is presented, outlining the fundamental concepts with a particular focus on data series representations and random forests. Section 2.4 presents a review of decision trees and Section 2.5 describes the core of ensemble methods and random forests. Chapter 3 presents the scientific approach, which includes an overview of the general research strategy employed in the individual studies, as well as a description of the philosophical assumption that motivates it. The evaluation framework, including performance metrics, validation methods, and significance tests, is presented and discussed in detail in Section 3.2. Chapter 4 summarizes the original contributions and findings of this dissertation, which includes both a theoretical and an empirical investigation of the presented algorithms in relation to the outlined research question and objectives. More specifically, Section 4.1 summarizes how high-dimensional and sparse representations can be handled in random forests. Section 4.2 recapitulates the results for the empirical investigation. This includes a presentation of the predictive and computational performance when evaluated on univariate, multivariate, and heterogeneous data series. Similarly, Section 4.2.3 presents and evaluates random forests for early prediction through a series of experiments. Finally, Chapter 5 concludes the dissertation, with an overarching discussion of the included studies in relation to the research question.

2. Random forests

In this chapter, supervised learning is introduced to provide a theoretical justification for the random forest algorithm. First, machine learning is introduced, followed by a discussion of traditional and data series representations and related work on data series classification. Second, to introduce the random forest algorithm, decision trees and ensemble methods are presented.

2.1 Machine learning

In the examples presented in Chapter 1, the motivating objective is to find a systematic way of predicting a phenomenon or classifying an object, often in the form of an output label, given a set of (input) variables related to the object or phenomenon. In machine learning, this goal is formulated as a supervised learning task, in which labeled inputs and outputs are given and the aim of supervised learning algorithm is to infer a model that is able to to predict the output label (used interchangeably with class) accurately for a previously unseen input example, based on the values of observed input and output variables.

Formally, in supervised learning, the learning algorithm is given a training set $\mathbf{Z}$ of training instances of the form $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ for some unknown target function $f : \mathcal{X} \to \mathcal{Y}$, where the target function denotes an ideal formula for mapping the (complete, possibly infinite) input space $\mathcal{X}$ to the (complete, finite) output space $\mathcal{Y}$ (in this thesis the output space is assumed to be categorical). Since only a sample of this data is given, the learning algorithm is required to use the (finite) training set $\mathbf{Z}$, where each $\mathbf{x}_i \in \mathbf{X}$ and each $\mathbf{y}_i \in \mathbf{Y}$ are taken from the set of input examples $\mathbf{X}$ and output (class) labels $\mathbf{Y}$, to pick a function $g : \mathcal{X} \to \mathcal{Y}$, from a set of hypotheses[1] $\mathcal{H} = \{h_1, \ldots, h_j\}$, such that $g \approx f$.

Statistically, the training data are random variables with their values taken jointly from the input and output space with respect to their joint probability distribution. Accordingly, the chosen hypothesis g, from the hypothesis space, is usually found by empirical risk minimization. In empirical risk minimization, a loss function $\mathcal{L}$ is employed to find the predictive model that best fits

[1]For instance, $\mathcal{H}$ could be all possible decision trees.

the training data. Assuming that the training set consists of a sample of *independent and identically distributed* instances, the resubstitution loss (training set performance) is defined as[1]:

$$E_{in}(h) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i), \tag{2.1}$$

where $h(\cdot)$ returns the prediction of $\mathbf{x}_i$ by the hypothesis and y_i is the true value of $\mathbf{x}_i$ as defined by f. Similarly, the generalization loss of a predictive model is the expected loss over the input space:

$$E_{out}(h) = \mathbb{E}_{\mathbf{x},\mathbf{y}}(\mathcal{L}(h(\mathbf{x}), \mathbf{y}). \tag{2.2}$$

Thus, E_{out} measures the loss of h in the underlying distribution. In this setting, given a learning algorithm $\mathbf{L}$ and a family of assumptions θ that instructs the execution of the algorithm, the application of said learning algorithm to the training set $\mathbf{L}(\theta, \mathbf{Z})$ results in a model $g \in \mathcal{H}$. The goal of finding a predictive model is consequently formulated as:

$$g = \arg\min_{\theta} E_{out}(\mathbf{L}(\theta, \mathbf{Z})), \tag{2.3}$$

which amounts to finding parameters that minimize the generalization performance according to the specified loss. Put differently, the primary goal of machine learning is to, given training data of finite size, identify a sufficiently good approximation, which is also reliable over the expected distribution of data from the complete input space. Since this requires E_{out} to be computed, it can, of course, not be solved exactly in practice. Instead, most machine learning algorithms aim at directly or indirectly optimize Equation (2.1) to learn the model. As a consequence, the resubstitution loss provides a poor estimate of the model's generalization performance. To know if a sufficient approximation is found, the generalization performance in Equation (2.2) is typically estimated from a hold-out sample which is drawn from the training set, but not used during training. Discussions related to such hold-out strategies are deferred to Section 3.2.2.

[1] $\mathcal{L}(y, y') = \mathbf{1}(y \neq y')$ where $\mathbf{1}(\cdot)$ return 1 if condition is true and 0 otherwise.

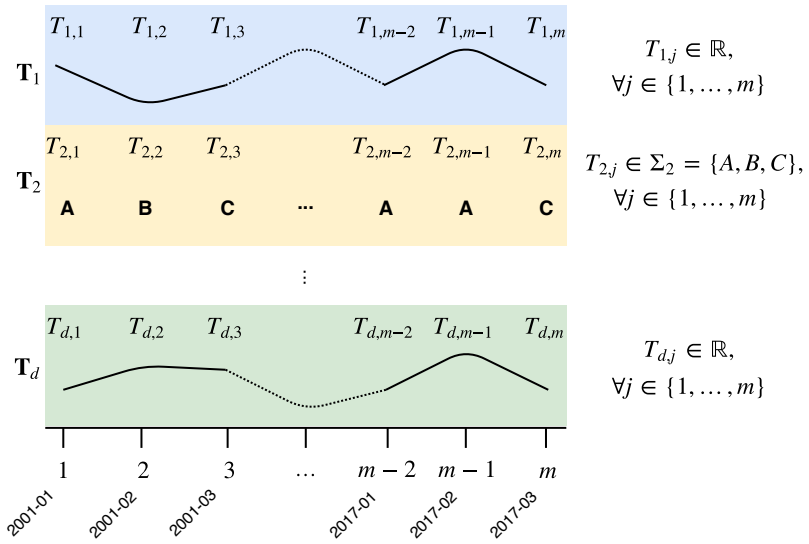

Figure 2.1: An example of a d-dimensional (heterogeneous) data series. The temporal axis consists of incremental year and month pairs and are shown on the horizontal axis. The temporal axis is shared by the all of the dimensions, i.e., $T_{k,j}$ is the value of the kth dimension at the jth time stamp. The values of the data series are shown on the vertical axis as $\mathcal{T} = \{\mathbf{T}_1,\ldots,\mathbf{T}_d\}$ associated to one of the domains in $\mathbf{B} = \{B_1,\ldots,B_d\}$. In the example, each value of dimension 1 and d are drawn from the real numbers, i.e., $T_1^j \in B_1$ and $T_{d,j} \in B_d$ where $B_1 = \mathbb{R}$ and $B_d = \mathbb{R}$. Similarly, the second dimension is drawn from the discrete domain $B_2 = \Sigma_2 = \{A,B,C\}$.

2.2 Data representations

In this section, representations used for machine learning algorithms are discussed. The first section introduces data series and the subsequent section briefly introduces traditional multidimensional representations.

2.2.1 Data series

A data series is an (often temporally) ordered series of data points that can be either homogeneous (all dimensions have the same value domain) or heterogeneous (the dimensions have different attribute domains). A data series can be either univariate (1-dimensional) or multivariate (d-dimensional). For a numeric data series, the domain $\mathbf{B} = \{\mathbb{R}_1,\ldots,\mathbb{R}_d\}$ is homogeneous and the value of each dimension is drawn from the real line. Moreover, the values of a data series have a logical ordering which can be, e.g., location, time or angle.

Definition 2.2.1. *A d-dimensional data series $\mathcal{T} = \{\mathbf{T}_1,\ldots,\mathbf{T}_d\}$ is a sequence*

of d variables, where $\mathbf{T}_k = (T_{k,1}, \ldots, T_{k,m})$ *with* $T_{k,j} \in \mathbb{R}$*. For simplicity, given a dimensionality of* $d = 1$*, let* $\mathcal{T} = (T_1, \ldots, T_m)$ *denote a univariate data series.*

The categorical analog to a (numeric) data series is a *discrete* data series. Similar to numeric data series data, a discrete data series is composed of a series of points along an axis, but instead of numerical points, the domain is categorical. The domain for each dimension of a discrete data series is defined by $B = \{\Sigma_1, \ldots, \Sigma_d\}$, where each Σ_k defines an alphabet for the kth dimension.

Definition 2.2.2. *A d-dimensional discrete data series* $\mathcal{T} = \{\mathbf{T}_1, \ldots, \mathbf{T}_d\}$ *is a set of d discrete data series, where* $\mathbf{T}_k = (T_{k,1}, \ldots, T_{k,m})$ *with* $T_{k,j} \in \Sigma_k$

An *heterogeneous* data series is a d-dimensional data series where the values of different dimensions can take on different domains, e.g., from Σ or $\mathbb{R}$. Henceforth, if not clear from context a data series refers to a data series with a numerical domain, a discrete data series refers to a data series with discrete domain and a heterogeneous data series refers to a data series with different domains. To clarify, $\mathcal{T}$ will be interchangeably used to refer to any type of data series and $\mathbf{x}_i \in \mathbf{X}$ will refer to a training set where each example is a data series, i.e., $\mathbf{x}_i = \mathcal{T}_i$. Figure 2.1 presents an example of a d-dimensional heterogeneous data series, where the values of the first and last dimension are drawn from the domain of real numbers, and the values of the second dimension are drawn from the simple alphabet of $\Sigma_2 = \{A, B, C\}$. In the example, the ordering is simply represented as $1, \ldots, m$ of year and month pairs.

2.2.2 Multidimensional data

Traditional supervised learning algorithm generally expects the input and output data in particular formats, e.g., only numeric values, only discrete values or a combination of both. For such multidimensional data, the $\mathbf{x}$ values are typically vectors of the form $\langle x_i^1, \ldots, x_i^m \rangle$, where each component x_i^j is taken from an attribute domain $a_j \in \mathbf{A}$ of either categorical or numerical values (let $\mathbf{X}^a$ represent all values of the ath attribute).

The process of identifying relevant attributes from raw data, which is not directly applicable to the learning algorithm, is often done manually by using expert knowledge. The processes of identifying such attributes is referred to as feature engineering and is often application specific. Discrete data series (e.g., text data or event sequences) can often simply be transformed to multidimensional representations by transforming the (discrete) data series to a training set that consists of the unique events as attributes and each value is an indication of the presence or absence of the event in the original data series. While this processes is not directly applicable to numerical data series, distritization techniques such as symbolic aggregate approximation (SAX) [Lin et al., 2007] can

be used to transform the numerical data series to a discrete data series. Other, more elaborate, transformation techniques are described in Section 2.3.2. Depending on the complexity of the transformation process, the resulting multidimensional representation can capture different properties of the data with different dimensionality. For instance, when transforming event sequences or text data to multidimensional representations the presence or absence of several thousands of unique events needs to be recorded, many of which is not present for many of the examples. This results in high-dimensional and sparse representations.

To remedy the issue of high-dimensionality and sparsity, several algorithms have been developed for dimensionality reduction [Cao et al., 2003]. Prominent techniques among these include principal component analysis [Wold et al., 1987] which captures attributes with high variability, independent component analysis which captures the weighed sum of independent non-Gaussian components [Comon, 1994] and multidimensional scaling which project distances matrices to a lower dimensionality while preserving the relative distances [Cox and Cox, 2000]. Although these methods work well for reducing the attribute dimensionality, the computational cost is considerable when training set contains a large numbers of attributes and examples. In text classification, where dimensionality is also a crucial issue, random indexing [Kanerva et al., 2000] is often applied to, e.g., compress bag-of-word representations.

2.3 Data series classification

Before proceeding to the next sections, in which the class of learning algorithms known as decision trees and random forests will be discussed in detail, related work on data series classification is briefly introduced here. Note that this is by no means an exhaustive enumeration of data series classifiers, instead it mainly focuses on the algorithms compared to in the empirical investigation presented in Section 4. For a complete overview of recent data series classification methods, the interested reader is referred to e.g., Bagnall et al. [2016].

2.3.1 Nearest neighbor methods

Nearest neighbor methods belong to a class of non-parametric predictive models known as proximity or prototype-based methods. As opposed to other learning algorithms, these methods does not learn a model but rather stores the training data and employ distance measures to find the instances closest to to the new example in order to make a prediction. Specifically, the k-nearest neighbor algorithm [see Hastie et al., 2009] find the majority output label among the k nearest neighbors:

$$g(\mathbf{x}) = \arg\max_{c \in \mathbf{Y}} \sum_{(\mathbf{x}', y') \in N_k(\mathbf{x}, \mathbf{Z})} \mathbf{1}(y' = c), \tag{2.4}$$

where $N_k(\mathbf{x}, \mathbf{Z})$ denotes the k nearest neighbors of $\mathbf{x}$ in $\mathbf{Z}$. Depending on application, domain or the nature of data series, various distance measures can be used to compare the similarity of data series.

Definition 2.3.1. *Given two data series $\mathcal{T}$ and $\mathcal{T}'$ with the same domain $\mathbf{B}$, the distance between their corresponding k:th dimension is dist$(\mathbf{T}_k, \mathbf{T}'_k)$.*

The most common distance measure for numeric data series is simply the Euclidean distance, defined as:

$$dist(\mathbf{T}_k, \mathbf{T}'_k) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (T_{k,i} - T_{k,i})^2}. \tag{2.5}$$

Since most approaches to the classification of data series are prototype-based, varying in their choice of distance measure, significant attention in the data series classification community has been given to improving the accuracy by employing different distance measures, e.g., elastic distance measures.

The benefit of elastic distance measures is that they are robust to misalignment and time warps, which allows for localized distortion, shifts, misalignments and data series of different length. In general, elastic distance measures include dynamic time warping or longest common subsequence [Maier, 1978] and variants, e.g., constraint-DTW [Sakoe and Chiba, 1978], Edit Distance on Real sequence (EDR) [Chen and Özsu, 2005], Edit distance with Real Penalty (ERP) [Chen and Ng, 2004]. Common for these is that they align data series with different lengths using dynamic programming. Specifically, DTW identifies the optimal match between two data series by allowing non-linearity in the distance calculation. Moreover, it has been shown that the generalization behavior of prototype methods can be significantly improved by bounding the DTW computation using a constraint [Ratanamahatana and Keogh, 2004]. In the context of classification, other important distance measures include weighted DTW [Jeong et al., 2011], time warp edit [Marteau, 2009] and move split merge [Stefan et al., 2013]. By combining several (11) elastic distance measures and weight their importance using their predictive performance in an ensemble, Bagnall et al. [2015] shows that great improvements in accuracy can be achieved. Unfortunately, the optimization protocol they employ requires significant computational resources [Bagnall et al., 2015].

Similarities between multivariate data series are, as pointed out earlier, not only captured by similarities between the individual dimensions but similarities can also be captured across dimensions. For example, a sudden increase

in one dimension can be followed by a sudden decrease in another. As such, extensions of the DTW algorithm for multivariate data series is non-trivial [Shokoohi-Yekta et al., 2015]. When classifying multivariate data series, several alternatives have thus been devised. The simplest and most commonly employed multivariate elastic distance is the cumulative distance of all univariate distances, which is equivalent to the distance found by concatenating the data series into a single univariate data series [Shokoohi-Yekta et al., 2015]. Another, more complex, multivariate similarity search preserves correlation between dimensions by bottom-up segmenting the data series and employing local similarities from principal component analysis similarity factors [Bankó and Abonyi, 2012].

2.3.2 Feature-based methods

As introduced in Section 1.2, another general class of predictive models relies on transforming the data series to a multidimensional data representation. More specifically, given a collection of data series $\mathbf{X} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$, a transformation function $\mathbf{t} : \mathbf{X} \to \mathbb{R}^{n \times d}$ and a learning algorithm $\mathbf{A}$, feature-based classifiers utilizes a transformation function and learns a predictive model from the transformed multidimensional dataset. For feature-based strategies the choice of algorithms is thus two-fold: the learning algorithm and transformation algorithm are jointly optimized. Feature-based predictive models are represented on the form:

$$g(\mathbf{x}) = g_{\mathbf{t}}(\mathbf{t}(\mathbf{x})) \tag{2.6}$$

where $g_{\mathbf{t}}$ is the predictive model induced by the transformation $\mathbf{t}$. For discrete data series, the simplest feature-based transformation is to, for each symbol in the alphabet, construct binary vectors that record whether or not a particular data series contains the symbol, these representations as discussed earlier disregards the ordering and often results in high-dimensional and sparse datasets.

Rodríguez et al. [2005] adopts an interval-based approach to identify a new dataset consisting of interval-attributes extracted from the data series. They employ two different kinds of binary attributes, the first consists of detecting if an interval has a mean lower than a threshold, and another that detects if an interval has a deviation lower than a threshold. The attributes are identified using boosting and a support vector machine is used as classifier. The algorithm presented by Rodríguez et al. [2005] act as motivation for two related interval-based transformation approaches, both of which investigates alternative approaches to (i) identify the intervals and (ii) to compute attribute values.

In addition to the interval averages and deviations, [Baydogan et al., 2013] propose, the rather heuristic, time series bag-of-words, which builds on in-

tervals which retains start and stop indices to allow for detecting temporal similarities. The algorithm has four main stages. The first stage consists in transforming each data series into new instances represented as average, standard deviation and slope for multiple intervals. In the second stage, the new representation is used in a random forest to generate probability estimates for each instance. The third stage recombines the probabilities and forms bags of patterns for each data series. These patterns consist of the class probabilities being discretized into bins and used for prediction in another random forest. While Baydogan et al. [2013] utilize random forest for the transformation, any machine learning algorithm can be used for this purpose and, similarly, for constructing the final predictive model.

Baydogan and Runger [2014] extends the time series bag-of-words framework to multivariate data series, denoted as a symbolic representation for multivariate time series (SMTS), by transforming the training set into a multidimensional dataset where the attribute space consists of temporal indexes and the values are the gradients between consecutive temporal values. Baydogan and Runger [2014] induces a random forest model from the transformed data with the motivation that the tree learners can capture trend information within and across dimensions. The leaf nodes of the decision trees are considered as symbols and the instances are re-represented as histograms of leaf node counts.

Similarly to SMTS, Baydogan and Runger [2015] propose learned pattern similarity (LPS), which uses the intervals as features rather than new instances for generating probability estimates. In LPS, the internal model (the probability estimator) is instead a regression model that is designed to detect correlations between intervals. The model randomly selects intervals and generates a new attribute matrix from which a random regression tree is constructed with a randomly selected attribute as a (regression) target value. The final transformation of each regression tree aggregate is the leaf node count to construct new instance representations. In the traditional formulation, a nearest-neighbor based method is used for classification, but any predictive model can be used [Baydogan and Runger, 2015].

Other examples of feature-based methods for data series classification include other interval-based features, such as statistical features [Nanopoulos et al., 2001; Rodríguez and Alonso, 2004], correlation structure features, distribution and entropy features [Fulcher and Jones, 2014], and wavelet features [Liu et al., 2009]. The primary drawback of approaches that consider feature representation a disjoint step from model induction is that the utility of the feature representation must be globally relevant. On the other hand, one benefit of this class of methods is that any traditional machine learning algorithm can be used [see Hills et al., 2014]. Moreover, since feature-based methods model data series separately from the predictive model, they are often trivial to extend

to multivariate data series [e.g., Baydogan and Runger, 2014, 2015].

2.3.3 Phase-independent shapelet-based methods

Another common class of data series classifiers is based on the concept of *data series subsequence*, which is a phase-independent local segment of a (longer) data series. These subsequences are referredd to as shapelets.

Definition 2.3.2. *A data series subsequence of the k:th dimension of a data series $\mathcal{T}$ is a sequence of l contiguous elements of $\mathbf{T}_k$, denoted as $\mathbf{T}_k^{s:s+l-1} = (T_{k,s}, \ldots, T_{k_s+l-1})$, where s is the starting index and l the length.*

To identify the utility of a shapelet, the distance between a data series and a shapelet must be identified, usually defined as the minimum distance achieved by sliding the shapelet along the data series to which it is compared and finding the best match [Ye and Keogh, 2009].

Definition 2.3.3. *Formally, given a 1-dimensional data series $\mathcal{S}$ (subsequence) of length l, and a d-dimensional data series $\mathcal{T}$, where each $\mathbf{T}_k$ is of length m and $l \leq m$, the data series subsequence distance between $\mathcal{S}$ and the kth dimension of $\mathcal{T}$ is the minimum distance between $\mathcal{S}$ and any subsequence of $\mathbf{T}_k$ of length l:*

$$Sdist(\mathcal{S}, \mathbf{T}_k) = \min_{s=1}^{m-l+1} \{dist(\mathcal{S}, \mathbf{T}_k^{s:s+l-1}\}. \tag{2.7}$$

where $dist(a,b)$ computes the distance between two sequences of the same length.

Moreover, a useful candidate is a shapelet that can partition the training set into two sets that have as pure a class distribution as possible based on the distance of the data series to the shapelet. Given a shapelet candidate the splitting criterion can be any measure of purity, e.g., entropy (see Equation (2.10)). The discovery of useful shapelet candidates has three main components: extraction, similarity between the shapelet and the data series, and utility, each affecting the computational and predictive performance of the resulting model. In the context of classification, shapelet-based approaches have predominately been used for the construction of *shapelet trees* (discussed in more detail in Section 2.4.3) [Ye and Keogh, 2009]. Since the extraction of shapelets is computationally costly, significant attention has been given to the efficient evaluation and extraction of shapelet shapelet candidates.

To improve the predictive performance of shapelet trees, feature-based methods (see Section 2.3.2), in which multidimensional representations are

produced, based on shapelets have been investigated [Hills et al., 2014; Wistuba et al., 2015]. In these approaches, each attribute of the multidimensional representation is the (minimum) distance from a shapelet to each data series in the database, i.e., in the input matrix each column is a shapelet, each row is a data series and each cell represents the (minimum) distance between them. Disconnecting the search for the shapelets from the model generation, which reduces the data series classification to an attribute selection (or generation) problem, enables the use of a wide range of efficient learning algorithms tailored for traditional data representations [Hills et al., 2014]. The original shapelet-based transformation (ST) approach identifies and selects the top k shapelets as attributes.

Since the original formulation of shapelet transformations extracts candidates based on multi-class information gain, an early abandonment of the computation of the entropy by upper-bounding requires the inspection of all class-wise permutations, which limits its usefulness. Hence, to speed up the identification of discriminatory shapelets, Bostrom and Bagnall [2015] proposes an alternative formulation of the shapelet transformation framework, in which the number of shapelets per class is balanced and the evaluation criterion is how well a shapelet discriminates between a particular class and all other classes. In the algorithm, all candidate shapelets for each data series are assessed in terms of their discriminatory power in a one vs. the rest setting, retaining in the final data representation a proportionate number of shapelets per class. Comparing the binary shapelet transform approach to the original transform, the former improves both predictive performance and computational efficiency [Bostrom and Bagnall, 2015].

In a related line of research, Wistuba et al. [2015] propose an alternative to identifying the top scored shapelets, which they refer to as the ultra-fast shapelet transform (UFS). The ultra-fast shapelet transform instead samples a large collection of shapelets at random, which avoids a search for the best shapelets. Using this formulation to transform the data series into high-dimensional representations (using thousands of shapelets), they show that the computational cost of the transformation can be significantly reduced compared to the traditional shapelet transform while still retaining competitive predictive performance.

To improve the predictive performance and not limit the selection of shapelets to those present in the training data, *shapelet learning* (LTS) has been introduced as an alternative exhaustive enumeration of shapelets. In shapelet learning, the shapelets are learned from the training data, while simultaneously minimizing both the training error, using a logistic loss function, and the minimum distance, using a soft minimum [Grabocka et al., 2014]. As opposed to other shapelet-based methods, shapelet learning does not restrict the shape-

lets to the subsequences found in the training data. Instead, the shapelets are initialized using clustering and candidates and logistic regression weights are jointly optimized for each class. By learning one logistic regression model per class, one drawback of shapelet learning is that the computational complexity increases with the number of classes in the training data [Grabocka et al., 2014].

2.3.4 Early prediction of time series

To make data series classifiers able to output predictions about phenomena early, i.e., by only inspecting a limited set of the temporal attribute $1\ldots,l$, where $l < m$, the goal of *early* time series prediction is to find a predictive model that can output a label with low error as early as possible [Xing et al., 2009]. In this definition, it is assumed that the class label is unchanged in the temporal dimension. Concretely, for a data series $\mathcal{T}$ of length m, the length-l prefix will be denoted by $\mathcal{T}^l$ and includes the data points from $1,\ldots,l$. Also, let $\mathbf{X}^l$ denote the training set of data series in the lth prefix space. Under this definition, the earliness of an early predictive model is defined as

$$Ea_{in}(e,h) = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n} e_g(\mathcal{T}_i) \tag{2.8}$$

where $e_g(\cdot)$ returns the earliest point in time for which g is allowed to output a prediction. In the early prediction framework, a trigger function e_g must thus be identified using a suitable learning algorithm. Note that a suitable predictive model must be able to output a prediction for a truncated time series, e.g., by keeping one model per time step. The goal of early time series prediction is, hence, to maximize the predictive performance and minimize the number of time instants required before prediction.

Algorithm 1 Prediction under the early prediction framework.

```
1: procedure PREDICT-EARLY(T, e, g)
2:     Initialize t to 1
3:     while true do
4:         if e_g(T^t) ≤ t then
5:             return g(T^t)
6:         end if
7:         t ← t + 1
8:     end while
9: end procedure
```

To illustrate, early classifiers typically follow the framework outlined in

Algorithm 1, the primary difference being the selection of the trigger function and the choice of algorithm for learning the predictive model. For example, trigger functions that rely on estimating the minimum prediction length (MPL), i.e., the earliest time for which a reliable prediction can be made, have been proposed [Xing et al., 2008, 2012]. These trigger functions are based on the l-length prefix-space of the data series, where, for a given data series, the algorithm finds the smallest l such that for any length $k < l \leq m$, the nearest neighbors in the $k, \ldots, l$ prefix are not different from the nearest neighbors in the full-length space (m). In this framework, the nearest neighbors in the lth prefix-space are used to make predictions if and only if the nearest neighbor has an MPL of at most l. Otherwise, the prediction is postponed until more data has arrived [Xing et al., 2012]. Other early prediction algorithms based on predictive models that output conditional probabilities have been investigated [see Dachraoui et al., 2015; Mori et al., 2016]. These approaches employ the precision of each class to estimate a point in time when an early prediction can be made. For instance, Mori et al. [2016] determine the MPL separately for each output label using the cross-validation error of the predictive models induced over a set of prefix-spaces. As a result, one drawback is that they require the induction of multiple predictive models, one for each time-step.

2.4 Decision trees

Before proceeding to a discussion of the random forest algorithm, which constructs an ensemble of randomized decision trees, this section introduces the decision tree algorithm for both multidimensional and data series representations. For multidimensional data, decision trees are one of the most common supervised learning algorithms. The popularity of decision trees is mainly due to their ability to produce both reliable and understandable predictive models. The induction of decision trees was independently described by Friedman [1976], Breiman et al. [1984], and Quinlan [1986].

The success of decision trees in the machine learning community can be attributed to four key benefits, compared to alternative algorithms. First, decision trees are non-parametric and can thus model arbitrarily complex learning tasks, given sufficient training data [Aggarwal, 2015]. Second, they support heterogeneous input vectors that consist of numerical, ordinal, and categorical values. Third, they are robust to noisy, irrelevant, and missing attribute values. Fourth, various forms of decision trees are the building block for many state-of-the-art predictive models, such as random forest [Breiman, 2001], boosting, and, in the context of data series classification, shapelet trees [Ye and Keogh, 2011].

2.4.1 Decision tree model

A decision tree is a tree-structured model where each node represents a partitioning of the input space. Although decision trees in the general case can be induced for multiple partitions at each node, the discussion here concerns binary decision trees, in which each internal node has exactly two children. In a decision tree, all internal nodes t are associated with a splitting criterion s_t, which divides the input space into disjoint subspaces. For binary decision trees, one of the subspaces is composed of the input examples for which the criterion is fulfilled and the second subspace is composed of the examples for which the criterion is not. Hence, the root node t_0 partitions $\mathbf{X}$ and, for each node, $\mathbf{X}_t$ is the partitioning (local training set) at node t. The terminal nodes of a decision tree are labeled with the most probable output $\hat{y}_t \in \mathbf{Y}_t$ of the output value, among the examples in $\mathbf{Z}_t$ reaching t. The most probable class label is expressed as $\hat{y}_t(\mathbf{x})$. The terminal nodes are in this way labeled with the best guess among the instances reaching the leaf. Formally, the prediction of an instance $\mathbf{x}$ is the label of the leaf that is reached by traversing the tree according to the splits s_t.

2.4.2 Induction of decision trees

Inducing the minimal optimal decision tree that partitions the training set and minimizes the resubstitution loss (e.g., the loss over the training set in Equation (2.1)) is NP-complete [Hyafil and Rivest, 1976]. For smaller datasets it is certainly possible to enumerate all possible decision trees and chose the smallest possible tree that explains the data, as instructed by the principle of Occam's Razor [Pearl, 1978; Quinlan, 1993], to avoid overfitting[1]. However, a more common strategy is to greedily learn a sufficiently good proxy by using one or more of several heuristics proposed in the literature [Breiman et al., 1984; Quinlan, 1993].

To heuristically and greedily induce a decision tree, Breiman et al. [1984] define an impurity measure $I(s,t)$ that evaluates the goodness of a split s at node t, with the assumption that the smaller the impurity value, the better the final leaf node $\hat{y}(\mathbf{x})$ predicts the examples $\mathbf{x} \in \mathbf{X}_t$ reaching the node. In this framework, the goal of the decision tree induction algorithm is to maximize the impurity decrease measure, i.e., the goal is to make each split as *pure* as possible. Breiman et al. [1984] define the impurity decrease for a binary split as

[1] *Overfitting* indicates that a model is too flexible and *underfitting* that it is not flexible enough.

$$Im(s,t) = I(t) - \frac{n_t^R}{n_t} I(t^R) - \frac{n_t^L}{n_t} I(t^L), \tag{2.9}$$

where n_t^R and n_t^L are the number of instances going to the right and left children of t respectively, n_t the number of instances reaching t, and $I(t)$ measures the goodness of a particular node t. The most popular impurity measures reach their minimum and maximum when the local class distribution is respectively maximally different or maximally similar[1] [Breiman et al., 1984]. Both the entropy [Quinlan, 1993], defined as

$$I_{entropy}(t) = -\sum_{c \in \mathbf{Y}} p(c|t) \log_2 p(c|t), \tag{2.10}$$

where $p(c|t)$ estimates the probability of c based on the class frequencies in $\mathbf{Y}_t$ and Gini index [Breiman et al., 1984], defined as

$$I_{gini}(t) = 1 - \sum_{c \in \mathbf{Y}} p(c|t)^2, \tag{2.11}$$

fulfill these criteria.

Algorithm 2 Generic decision tree induction algorithm.

```
1:  procedure InduceDecisionTree(Z)
2:      Create a node t
3:      if Z is pure or other stopping criteria is met then
4:          ŷ_t = dominant class label
5:      else
6:          Find most informative split: s' = arg max_s Im(s,t)
7:          Partition Z_t according to s' as Z_{t_L} and Z_{t_R}
8:          for both L and R as v do
9:              t_v ← InduceDecisionTree(Z_v)
10:             Attach t_v to the corresponding branch of t
11:         end for
12:     end if
13:     Return t
14: end procedure
```

Based on the impurity measure in Equation (2.9), a greedy algorithm for inducing a decision tree is described in Algorithm 2. In the algorithm, several details require attention. Typically, the assignment function on Line 4 predicts the maximally prevalent class in $\mathbf{Z}_t$,

[1] Minimal at $I(\{1,\ldots,0\}),\ldots,I(\{0,\ldots,1\})$ and maximal at $I(\{\frac{1}{|\mathbf{Y}|},\ldots,\frac{1}{|\mathbf{Y}|}\})$.

$$\hat{y}_t(\mathbf{x}) = \arg\max_{c \in \mathbf{Y}_t} p(c|t).$$

Since the resubstitution error of a decision tree is a constantly decreasing function of the tree depth[1], it appears that growing a decision tree to maximal depth is preferable. However, since increasing the complexity of the model leads to decision trees that capture noise in the training data, it generally leads to overfitting. There is thus a trade-off between too deep and too shallow trees. In Algorithm 2, this trade-off is expressed as a pruning criterion on Line 3. The most common pruning strategies are to stop the induction if the number of examples reaching the node is below some constant, if the depth of the tree is greater than some constant, or if the decrease in impurity is less than some constant. Since pruning has been shown to be detrimental to the performance of random forests, extensive discussion of the topic is omitted here.

Algorithm 3 Algorithm for finding the best split.

```
 1: procedure FINDBESTSPLIT(Z_t)
 2:     Initialize Δ to ∞
 3:     for each attribute a ∈ A do
 4:         Find the best binary split s_a in X_t^a
 5:         if Im(s_a, t) < Δ then
 6:             Δ = Im(s_a, t)
 7:             s' = s_a
 8:         end if
 9:     end for
10:     Return s'
11: end procedure
```

Finally, the choice of splitting point that maximizes the impurity decrease, on Line 6, is typically found according to Algorithm 3. Hence, for each attribute $a \in \mathbf{A}$, one should find the best binary partition of the examples $\mathbf{X}_t$ reaching the node and select the attribute that maximizes the impurity decrease. For categorical features, finding a binary partitioning amounts to selecting the attribute value $v \in \mathbf{X}^a$ that decreases the impurity the most if split upon. Simplified, a split is created that partitions, in a binary fashion, the input space into two disjoint sets: one for those examples where $x_i^j = v$ is true and one for those where it is false Quinlan [1993].

Similarly, given a numerical attribute $\mathbf{X}^a$, the best split s'_a of $\mathbf{X}^a$ is the binary partitioning of $\mathbf{X}_t$ into the subsets

[1] A decision tree with as many leaf nodes as training instances will, clearly, have a training error of 0. See Quinlan [1993] for a complete discussion.

$$\mathbf{X}_{t_L}^{v} = \{(\mathbf{x}, y) | (\mathbf{x}, y) \in \mathbf{Z}_{\mathbf{t}}, x^a \leq v\} \tag{2.12}$$
$$\mathbf{X}_{t_R}^{v} = \{(\mathbf{x}, y) | (\mathbf{x}, y) \in \mathbf{Z}_{\mathbf{t}}, x^a > v\} \tag{2.13}$$

where v is the numerical threshold of the split. The standard algorithm for finding the threshold v is to order the examples according to their attribute values and choose the mid-point ($\frac{v^k + v^{k+1}}{2}$) between two examples of different output labels so that the impurity decrease is maximized [Quinlan, 1986].

2.4.3 Shapelet-based decision trees

For data series classification, the most prominent decision tree induction algorithm is based on shapelets, i.e., class discriminatory subsequences referred to as shapelet trees (FST). The algorithm was first proposed by Ye and Keogh [2009, 2011] and subsequently extended by Hills et al. [2014] and Rakthanmanon and Keogh [2013]. In the traditional decision tree framework, a shapelet tree is a decision tree generated by combining feature extraction and the top-down recursive decision tree induction algorithm. Given a training set $\mathbf{Z}$ of (univariate) data series $\mathbf{X} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ and categorical output labels, the goal of the shapelet tree induction algorithm is to produce a decision tree where each internal node is represented by a shapelet $\mathcal{S}_t$ and a distance threshold v_t and each leaf node is associated with a class label $\hat{y}_t$. To induce such decision trees, Algorithm 3, which finds the best split, is substituted for Algorithm 4, which finds the best shapelet and threshold, in the generic tree induction algorithm.

In Algorithm 4, all shapelets of all sizes are extracted on Line 3, where extract$(\cdot)$ finds all shapelets of a particular data series. A procedure similar to Algorithm 3 is then utilized to, for each shapelet $\mathcal{S}$, find the best binary split based on the distances from the data series in the local training set. Line 6 of Algorithm 4 is thus used to compute the distances between $\mathcal{S}$ and all data series in the local training set. The resulting distance list $\mathbf{D}$, is subsequently discretized, similar to the way numerical attributes are handled in the decision trees outlined earlier, i.e., by sorting the distances and at each possible split point v' evaluating the impurity measure for a binary split, selecting the split that maximizes the impurity decrease. A valid split point is thus the mean of two consecutive distances for which the associated instances are labeled differently. Given a candidate shapelet split $s_{\mathcal{S}}$, the local training set $\mathbf{X}_t$ is partitioned into two sets:

Algorithm 4 Algorithm for finding the best shapelet split.

procedure FINDBESTSHAPELET($\mathbf{Z}_t$)
    Initialize Δ to ∞
    Initialize $\mathbf{S} = \{\mathcal{S} | \mathcal{S} \in \text{extract}(\mathcal{T}) \in \mathbf{X}\}$
    for each shapelet $\mathcal{S} \in \mathbf{S}$ **do**
        Compute the distance between the shapelet and each data series:

$$\mathbf{D} = \{Sdist(\mathcal{S}, \mathcal{T}) | \mathcal{T} \in \mathbf{X}\}$$

        Find the best binary split $s_{\mathcal{S}}$ based on $\mathbf{D}$
        if $Im(s_{\mathcal{S}}, t) < \Delta$ **then**
            $\Delta = Im(s_{\mathcal{S}}, t)$
            $s' = s_a$
        end if
    end for
    Return s'
end procedure

$$\mathbf{X}^v_{t_L} = \{\mathcal{T} : \mathcal{T} \in \mathbf{X}_t, Sdist(\mathcal{S}, \mathcal{T}) \leq v'\} \tag{2.14}$$
$$\mathbf{X}^v_{t_R} = \{\mathcal{T} : \mathcal{T} \in \mathbf{X}_t, Sdist(\mathcal{S}, \mathcal{T}) > v'\}. \tag{2.15}$$

The selected split hence results in a partitioning of the data series into two groups: one group for data series with minimum distance $\leq v'$ and one group with minimum distance $> v'$.

Since the number of possible shapelets (i.e., the attribute space) is clearly huge[1], various methods for speeding up the distance search have been investigated and evaluated in different contexts, for example, by early abandonment of the distance computations and lower-bounding the scoring function [Ye and Keogh, 2009], or by employing different, less costly, score functions based on the analysis of variance [Hills et al., 2014; Lines et al., 2012]. To speed up the shapelet tree algorithm, Rakthanmanon and Keogh [2013] proposes an extension, Fast Shapelets (`SAX`).

The fast shapelet algorithm discretizes the shapelets using SAX [Lin et al., 2007], which reduces the dimension of the data series using piece-wise aggregate approximation (PAA) [Keogh et al., 2001] and allocates the values to normally distributed bins. By forming dictionaries of SAX representations

[1]In fact, given a dataset of data series of the same length m, the total number of shapelets is $O(m^2 n)$. Since checking the utility of a single shapelet is $O(mn)$, the overall time complexity is $O(m^3 n^2)$.

for the shapelets and reducing their dimensionality through masking, word-frequency histograms are constructed for each target label. The algorithm then selects the top scored SAX-shapelets by comparing the discriminative power of the histograms. Finally, the SAX representation is remapped to the original shapelets and reassessed using information gain. In addition to these techniques for speeding up the exhaustive search, several other approaches have been proposed in the literature. Notably, methods for trading time complexity for memory consumption, while finding the optimal match, have been explored [Mueen et al., 2011].

An early and very similar decision tree induction algorithm for data series patterns was described by Geurts [2001]. In this algorithm, decision trees are constructed, similar to shapelet trees, using data series patterns and distance thresholds as internal nodes. However, instead of constructing patterns from the original data series and extracting patterns directly, the data series is discretized by a piecewise model and discontinuity points are used to define interesting patterns. The discretization process used by Geurts [2001] recursively constructs piecewise constant models based on a regression tree. By choosing the number of segments (leaf nodes in the regression trees) during the discretization process, the complexity of the resulting model can be tuned. Finally, the decision tree induction algorithm extracts a candidate pattern during node splitting from one randomly selected data series per class and finds the optimal threshold. The investigation of a single pattern per class and node is motivated by the computational requirements of investigating all possible patterns, even after discretization. Ye and Keogh [2009] partly mitigates these computational issues by employing an admissible pruning technique that upper-bounds the information gain [Ye and Keogh, 2009].

2.5 Ensembles

In this section, the attention is turned to ensemble methods, which are revisited to motivate and provide the necessary theoretical details for understanding random forests. In machine learning, ensembles are models that combine the predictions of a set of individual base-models. In this section, strategies for constructing ensemble members and combining their output are introduced and discussed in relation to the bias–variance trade-off. The primary focus of this discussion is to motivate the utility of the random forests algorithm.

Ensemble methods in supervised learning are motivated by the fact that different predictive models produce different outputs for the same inputs. Such differences may be due to the assumptions and characteristics of the learning algorithm or artifacts in the training data. An ensemble method is thus an approach for increasing the predictive performance by combining the outputs

from a set of induced hypotheses (referred to as base-models) $\{h_1, \ldots, h_p\} \in \mathcal{H}$[1]. Intuitively, it can be clearly seen that a necessary condition for an ensemble to be more accurate than any of its constituent hypotheses is that the base-models are more accurate than random guessing, and diverse. This intuition is popularly captured in Condorcet's jury theorem from the field of political science. The theorem states that if a group of independent voters that are more likely to be right than wrong wishes to reach a decision by majority vote, then inviting more voters increases the probability of the decision to be correct [De Condorcet, 1785]. Put differently, given that the base-models are more accurate than random, and that errors are somewhat independently distributed, an ensemble outperforms the mean accuracy of its constituent base-models [Hansen and Salamon, 1990]. This intuition is captured by the bias–variance trade-off.

2.5.1 Bias–variance trade-off

The bias–variance trade-off decomposes the error of a predictive model into three parts: its *bias*, *variance* and irreducible *noise*. The bias of a predictive model is related to systematic errors, due to inherent assumptions about the decision boundary that a classifier produces. For example, a linear predictive model can clearly not capture non-linear decision boundaries and thus has an inherent bias. A predictive model with a *high bias* will, therefore, make consistently incorrect predictions, even when the learning process employs different training data. On the other hand, in cases where variations in the choice of training data lead to different predictive models, the induced predictive models will make inconsistent predictions for the same test instance over different hypotheses and is thus variable. In this way, the bias–variance trade-off provides a useful tool for diagnosing and understanding the predictive performance of predictive models.

To show the trade-off between bias and variance, Figure 2.2 illustrates the decision boundary of predictive models induced using two different learning algorithms with examples of two different classes illustrated as blue and yellow circles. The gray lines represent predictive models[2] induced from random subsets of the training set, and the thick red lines represent the predictions based on their combinations. Clearly, none of the linear classifiers (right) can capture the complex (true) non-linear decision boundary[3], which translates into high bias and low variance (due to the limited model complexity). On the other hand, individually the complex deep decision trees (left) clearly overfit

[1]The hypothesis set can contain models induced by different algorithms.

[2]Decision trees to the left and linear support vector machine to the right.

[3]Note the intrinsic error, evident in the class overlap.

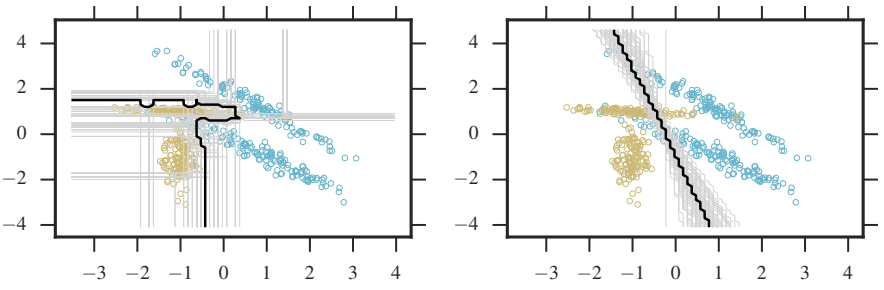

Figure 2.2: Decision boundary for a synthetic binary classification task. The red line shows the decision produced by combining the gray lines. The left figure shows the combined decision boundary of decision trees (high variance) and the right shows the decision boundary of linear support vector machines (low variance).

to random noise in the training data, which negatively impacts the generalization performance. As a result of high variability, however, the error of these models can be reduced in the combined model, which results in improved generalization performance.

More concretely, bias is the squared difference between the true conditional probability of $\mathbf{x}_i$'s being y_i and the prediction produced by the hypothesis. Consequently, the bias is small when the predictive model is consistently correct. Variance is the variation of the predictions of the chosen hypothesis. In contrast to bias, variance measures how inconsistent the output probabilities are and not whether they are correct or incorrect [Boström, 2012; Manning et al., 2008]. As a result, an effective ensemble method improves the predictive performance by reducing the bias and/or the variance of the combined models. By appropriately choosing the base-models of an ensemble with specific bias-variance properties and combining them in a suitable way, both bias and variance can often be reduced from those of any single model, thus producing an ensemble that is more effective than any of its constituent members [Aggarwal, 2015].

Dietterich [2000a] provides a complement to the bias–variance analysis, in which fundamental reasons intuitively explain why ensembles often perform better than single models from three points of view: *statistical*, *computational* and *representational*. First, if the size of the training set is small in relation to the hypothesis space $\mathcal{H}$, a learning algorithm can typically identify several predictive models that have the same performance on the training data. Provided that the predictions made by these models are somewhat uncorrelated, averaging these models reduces the risk of choosing the wrong hypothesis (see Figure 2.2 (left)). Second, since many learning algorithms are greedy and rely on searches that may end up stuck in local optima, an ensemble can combine

several models from different starting points which might provide a better approximation of the true unknown function and by employing smaller subsets of the data to reduce the computational burden. The final, representational, reason is that the true function often cannot be represented by any of the hypotheses in $\mathcal{H}$. By combining several models it might be possible to expand the space of representable functions in order to model the true function better.

2.5.2 Ensemble methods

The core principle behind ensemble methods based on randomization (this discussion is primarily focused on variance reduction ensembles such as random forests) is to introduce (random) perturbations into the learning procedure and thus generate several, slightly different, models from a single training set. As introduced, the predictions of those models are then combined to form the prediction of the ensemble, which (hopefully) reduces the variance, without increasing the bias too much. Many strategies for constructing ensembles have been developed and studied to increase the predictive performance by reducing the bias, variance, or a combination of both. Generally, ensemble methods explicitly introduce randomness into the learning algorithm or exploit randomized versions of the training set in each run of the learning algorithm to produce more or less diversified predictive models. As a result, variance is introduced both in the training set and model randomization, resulting in more variable base-models and to a lesser extent, depending on the amount of randomness, an increased bias.

Strategies for randomization

One of the most common strategies for generating multiple base-models is to manipulate the training set. In this strategy, the algorithm is run several times, each time with a different random (sub)set of training instances. For example, *bagging* introduces randomization by building each base-model from a bootstrap sample of training instances drawn with replacement from the original training set. Ensemble methods that rely on manipulated training sets have been shown to work especially well for unstable learning algorithms [Breiman, 1996; Dietterich, 2000b], i.e., algorithms with low bias and complex decision boundaries. Related to bagging, which draws instances with equal probability, wagging is another example of training set manipulation that draws examples according to stochastically assigned weights [Bauer and Kohavi, 1999]. The reader interested in a complete review of ensemble methods based on manipulating the training set is referred to Kuncheva [2004].

The second common approach for constructing ensembles is to manipulate the input attributes employed by the learning algorithm. For multidimensional

data, for example, reducing the number of input attributes employed by the algorithm during base-model construction lowers the computational effects of dimensionality, and thus improves the computational performance of induction [Friedman, 1997]. From a bias–variance perspective, another important effect of utilizing attribute selection or sampling methods is that it facilitates the construction of more diversified base-models [Kuncheva et al., 2001]. Random attribute selection is perhaps the most well-known and successful approach for manipulating the input attribute set [Ho, 1998], which also has the added benefit or reducing model construction costs.

Finally, another general strategy for constructing ensembles is to inject randomness into the learning procedure of the algorithm. For example, randomness has, with success, been introduced in neural networks by setting the initial weights randomly [Pollack, 1990] or by randomly removing links between layers [Johansson et al., 2013; Srivastava et al., 2014]. Randomization has also been applied to ensembles of decision trees. In this context, Dietterich [2000b] investigates ensembles of C4.5 [Quinlan, 1986] decision trees where a random top-ranked attribute is selected, similar to the random tree algorithm by Amit et al. [1997]. Ensembles of randomized support vector machines have also been investigated [Raviv and Intrator, 1996; Valentini and Dietterich, 2004]. By injecting randomness both in the selection of the decision tree attributes and the training samples, Breiman [2001] introduces random forests. Section 2.6 explores random forests in greater detail.

Combining models

Given a set of base models $g = \mathbf{H} = \{h_1 \ldots, h_p\}$ created by a suitable ensemble approach, the predictions must be combined in a way so that the expected generalization performance improves over the average of the individual models. In general there are two main approaches for combining the base models, usually referred to as (supervised) meta-learning strategies or simple fusion strategies [Rokach, 2010]. An example of a meta-learning approach that has proven successful is stacking, in which the outputs of several predictive models are provided as inputs to another learning algorithm. The stacked learning algorithm then induces a function that maps the outputs of the base models to the target label [Wolpert, 1992]. Since such approaches require additional training data and are, in some cases, prone to overfitting, unsupervised combination strategies are more common [Zhou, 2012].

In classification, predictions are usually combined by considering the ensemble as a committee, in which the label that receives a majority of the *votes* is the final prediction. Thus, majority voting assigns the output label that a majority of the base models predict:

$$\mathbf{H}(\mathbf{x}) = \arg\max_{c \in \mathbf{Y}} \sum_{k=1}^{p} \mathbf{1}(h_k(\mathbf{x}) = c), \tag{2.16}$$

with the rationale that the majority vote minimizes the average zero–one loss with respect to predictions made by the individual models. Ties can be handled by selecting a random label among the tied label or by, in cases where the individual models output conditional class probability estimates, performing distributional summation:

$$\mathbf{H}(\mathbf{x}) = \arg\max_{c \in \mathbf{Y}} \frac{1}{p} \sum_{k=1}^{p} \hat{P}_{h_k}(y = c|\mathbf{x}) \tag{2.17}$$

Several variants of distributional summation have been investigated, for instance, weighting the conditional probabilities by the base-model performance [Zhou, 2012] or by calibrating the probabilities [Boström, 2012]. Empirically, soft voting and majority voting perform similarly [Breiman, 1996], but soft voting has the advantage of providing more well-calibrated probability estimates which can be useful in situations where certainty about predictions is important. Furthermore, soft voting has the added benefit of being directly applicable to the bias–variance decomposition. Finally, to take into consideration the importance of the different base-model, majority voting in Equation (2.16) can be modified to weight the importance of each base-model, $\alpha_1, \ldots, \alpha_p$, where the weight of each predictive model can be set according to the predictive performance on a validation set, e.g., using out-of-bag examples, such as $\alpha_k = \frac{1-e_k}{\sum_{j=1}^{p} 1-e_j}$, where e_k is the error of the kth base-model. Rokach [2010] provides an extensive review of ensemble combination and creation strategies.

2.6 Random forests

As seen in Figure 2.2 (left), decision trees appear to be ideal candidates for ensembles since they usually have low bias and high variance. As noted earlier, random forests [Breiman, 2001] form a family of predictive models that consists of constructing an ensemble of (randomized) decision trees. Various variants of the same principle have been investigated but they chiefly differ in the way randomness is injected both in the training set and in the decision tree learning procedure, while simultaneously maintaining the low bias of the individual models.

In the context of decision tree ensembles, Dietterich [2000b] considers the construction of randomized decision trees where, in each node, a random split is selected among the top 20 best splits. This approach has been shown to improve the predictive performance over simply bagging fully grown decision

trees, especially in noisy settings. To reduce the computational burden of evaluating all splittings and selecting a random attribute among the top attributes, Amit et al. [1997] propose a randomized tree induction algorithm. This algorithm searches for the best splitting among a subset (i.e., altering Algorithm 3 to iterate over a subset) of l attributes at each node. In cases where there are many attributes, this has been shown to reduce the cost of tree induction significantly [Amit et al., 1997]. From a bias–variance point of view, if there are many correlated attributes, selecting attributes at random will make the trees structurally different, without increasing the bias. At the same time, the increase in variance can be canceled by averaging. By combining both bagging and random attribute subspaces, Breiman [2001] introduces the Random Forest and shows that injecting randomness both in the training set and during tree construction yields one of the most effective (and efficient) general purpose predictive models. In fact, a large-scale empirical investigation shows that among the 179 classifiers compared over 121 datasets, random forests, with the number of inspected attributes inspected at each node l optimized, significantly outperforms the alternatives [Fernández-Delgado et al., 2014].

Since its inception, variants of the random forest algorithm have been investigated to improve the predictive, computational or representational performance. To reduce the computational cost, Cutler and Zhao [2001] proposes a random forest variant, the "perfectly random tree ensemble" (PERT), in which the trees are fully grown without regard to the target variable, using random attributes and numerical splitting points. An empirical investigation shows that PERT performs similarly to the original formulation in terms of predictive performance while being significantly faster at induction, since there are no impurity computations. Related to perfectly random tree ensembles, extremely randomized trees (ETs) have been introduced. "Extremely randomized trees," somewhat similar to PERT, combines random attribute selection with discretization thresholds drawn at random from a uniform distribution [Geurts et al., 2006].

In a related direction, rotation forests have been introduced to simultaneously decrease tree correlation and increase individual tree accuracy [Rodriguez et al., 2006]. As with bagging, a rotation forest generates each tree from a bootstrap replica of the original training set. However, to decrease correlation among the resulting trees, the attributes are partitioned into subsets for which principal component analysis is performed and used to construct new attribute projections. In comparison, rotation forests perform better than bagging, boosting, or random forests, for a variety of tasks [Kuncheva and Rodríguez, 2007]. In a related direction, random forests consisting of decision trees that separate the attribute space by randomly oriented hyper-planes have been investigated. These random forest models find oblique splittings using

linear discriminant models in the nodes. In empirical investigations, oblique random forests have been shown to compare favorably to traditional random forests in terms of predictive performance Menze et al. [2011].

For data series classification, Deng et al. [2013] introduce a time series forest based on a random selection of interval-features, which consists of averages, standard deviations, and slopes. To reduce the computational burden of investigating all possible intervals, a limited subset is considered within a random forest framework. Hence, in this formulation, decision trees are constructed by randomly considering the interval features, and then combined into an ensemble (constructed without bootstrap aggregates). To improve the speed of the tree induction, the decision tree algorithm only considers a fixed number of pre-defined attribute split points instead of sorting the attribute values. To improve the splitting criterion, a heuristic to differentiate between interval features with equal entropy is introduced that selects the feature for which the largest margin is achieved. Although these simple interval estimates produce forests with high predictive performance, they fail to identify phase-independent and localized patterns.

Random forests have evolved during the last decades as a result of developments in both the theoretical analysis of ensemble methods and developments in decision trees, to now have become one of the most firmly established (baseline) predictive model for almost any application or domain. Despite this widespread interest and practical use, the theoretical properties of random forests are still not particularly well understood. Several studies to prove their consistency, i.e., that the error of the model converges to an optimal estimator as the training set grows infinitely large, have been proposed [see Biau, 2012; Denil et al., 2014; Scornet et al., 2015]. Nevertheless, the random forest algorithm has seen widespread empirical success, performing well for almost any general prediction task [Fernández-Delgado et al., 2014].

3. Scientific approach

In this chapter, the methodological and philosophical assumptions underlying this thesis will be introduced to help explain the research strategy and experimental method. After outlining the general research strategy, Section 3.2 presents a framework for evaluating predictive models, using quantitative performance metrics. To ensure that the obtained measures generalize to new observations, approaches that allow for estimating the generalization performance will be introduced in Section 3.2.2, together with statistical hypothesis testing in Section 3.3. The major part of this chapter is based on Montgomery [2008], Demšar [2006], García and Herrera [2008], and Alpaydin [2014].

3.1 Research strategy

The research strategy of this dissertation, similar to most research in machine learning, is predominantly quantitative in nature. This means that the research strategy is based on metrics that evaluate the performance of predictive models while investigating the effect of changing various factors, e.g., the effect of employing various learning algorithms, or processes to induce predictive models. Relatively similar to other branches of science and engineering, research questions in machine learning are usually addressed by deductive reasoning [Alpaydin, 2014]. With this kind of reasoning, a positivist paradigm is often assumed, which involves establishing cause-and-effect relations (using statistical tests) between empirical evidence and (falsifiable) theories or hypotheses [Montgomery, 2008]. As opposed to interpretivist reasoning, the positivist framework assumes that information from measurements, interpreted through statistical and logical reasoning, forms the basis for knowledge through constant facts found in empirical or theoretical evidence. Thus, the main questions related to research design using this framework concern how measurements are collected, analyzed, and interpreted, to allow for reliable conclusions to be drawn.

In the positivist and empirical framework, an experimental strategy is typically employed to reliably produce empirical evidence, which provides an understanding of the cause-and-effect relations that govern the changes in an output based on modifications to the inputs of a process [Alpaydin, 2014; Mont-

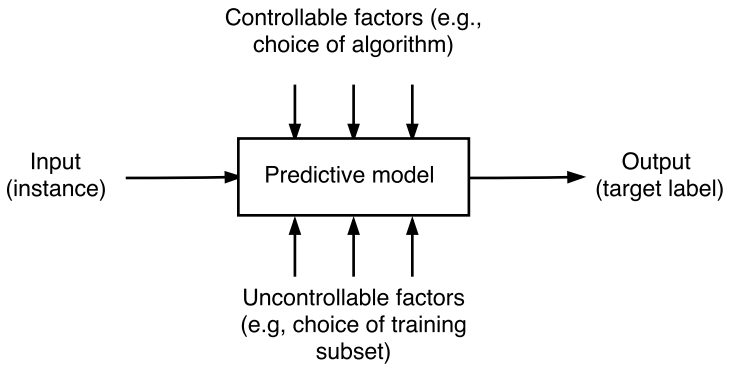

Figure 3.1: onstructing predictive models in a machine learning experiment. The predictive model generates a target label given a training instance. The resulting label is affected by controllable and uncontrollable factors. This figure is adapted from Montgomery [2008]

gomery, 2008]. Since the research question of this dissertation resolves around how to create efficient and effective random forests, and many of the included studies aim to establish the factors that affect the predictive performance of such models, a quantitative and experimental method is employed here. Hence, approaches for constructing efficient and effective random forests are explored in various ways, where the presented studies sequentially modify and introduce various factors, while investigating their effects. Consequently, to address the objectives, the research strategy is also, in one sense, characterized as being exploratory, i.e., seeking novel solutions. In fact, many equally interesting algorithmic choices could perhaps have been investigated and evaluated.

3.1.1 Machine learning experiments

A machine learning experiment is a series of tests in which changes to the induction process that produces a predictive model (Figure 3.1) affect some chosen performance metrics, typically based on the output predictions. Some examples of factors that, generally, affect the performance of a predictive model include the choice of learning algorithm and its hyper-parameters, input representations, and training data. More specifically, for an experiment, one typically differentiates between two types of factors commonly known as *controllable* and *uncontrollable* factors. As their names imply, controllable factors are those that can be altered (e.g., the choice of learning algorithm), whereas uncontrollable factors are those that cannot (e.g., randomness in the learning algorithm or training data). As seen in Figure 3.1, the output of the employed

predictive model is subsequently used to generate a response variable (i.e., a prediction). In a traditional supervised setting, a number of datasets, for which experts have assigned output labels, are used to generate the response for two or more predictive models (e.g., induced using different algorithms) with the aim of detecting (significant) differences in the chosen performance metrics. To detect such differences, a null-hypothesis is constructed, which states that there are no differences in performance (with regard to the selected performance metric) between a set of differently induced predictive models. The goal of an experiment is thus to identify what (controllable) factors affect the generated response variable (output), while at the same time reduce the impact of uncontrollable factors. By having control of the factors that influence the output, an experiment provides measurements that allow for reliable conclusions regarding the null-hypothesis to be reached [Alpaydin, 2014].

Although experiments can be designed in a multitude of ways, the most common designs include the following: best guess, one-factor-at-a-time, and factorial design [Montgomery, 2008]. Similar to an exploratory research strategy, the best guess design investigates the controllable factors in a less systematic way, letting intuition and heuristic decisions guide the process. To answer the overall research question of this thesis a best guess based approach thus is employed, where factors, in the respective studies, are investigated in a way that sequentially guides the decision of what to investigate next. In fact, the investigated controllable factors, related to the research question and objectives, presented in this thesis range from algorithm hyper-parameters (Papers **I** and **II**) and algorithmic design decisions (Papers **III**, **VII** and **VIII**) to the choice (and implementation) of learning algorithm (Papers **IV**, **V** and **VI**). In addition to the best guess approach employed to provide an answer to the overall research question, one-factor-at-a-time and factorial designs are employed in the individual studies to detect significant differences in predictive performance between alternative algorithms, hyper-parameters, and data representations in order to answer the sub-questions.

As mentioned, the evaluation of different strategies is conducted in an empirical and experimental setting, where the relative performance of several predictive models is evaluated using metrics of effectiveness and efficiency. To be able to address the research question of the dissertation, *efficient* and *effective* must defined. In this thesis, effectiveness is quantified using performance metrics computed from the output of the predictive models and compared to a reference standard. Efficiency, on the other hand, is not related to the predictive performance, and is as such defined as training and prediction time.

To ensure that the proper metrics of efficiency and effectiveness are employed, it is important for an experiment to ensure that the produced results are valid. Validity (internal validity) relates to credibility, and primarily concerns

the ability of an experiment to measure what it set out to measure, e.g., if the intent is to compare the cost of learning algorithms, CPU time might be a more suitable measure than wall clock time; whereas wall clock time might be more suitable if the intent is to explore the utility of implementations. Two aspects of an experiment closely related to validity are the concepts of *generalizability* (external validity) and *reliability*. First, the generalizability of a machine learning experiment concerns the extent to which the conclusions drawn from empirical data about the learning algorithms can be trusted and generalized to new settings. To ensure the external validity of an experiment, the results must be significant (see Section 3.3) and based on sufficient empirical data (see Section 3.2.2). Second, an experiment is considered to be repeatable if the experiment for collecting the empirical evidence can be reliably repeated.

For machine learning experiments, the internal validity greatly depends on the correct choice (for the domain) of performance metric (discussed in Section 3.2.1), e.g., if the class distribution is unbalanced, skewed between training and testing, or the miss-classification cost is unknown, the area under the ROC curve is typically more useful than the accuracy. Furthermore, the reliability of such experiments fundamentally depends on the size of the training data, where more data results in more reliable estimates of the generalization error. By having access to more data, the statistical evidence found using significance tests (discussed in Section 3.3), ensures that the conclusions that are drawn from the experiment will extend to new settings. Finally, the generalizability of a machine learning experiment concerns how the observed performance is expected to be maintained when the predictive model is applied to previously unseen data.

3.2 Evaluating predictive models

In this section, methods for ensuring the validity, reliability, and generalizability of machine learning experiments are discussed. This discussion concerns both the metrics for evaluating the computational and predictive performance and the methods for estimating the generalization behavior of the predictive models. Finally, this section concludes with a discussion concerning the statistical hypothesis testing which is used to identify statistically significant differences in the performance estimates when employing different factors that influence the resulting predictive model, e.g., the learning algorithm.

3.2.1 Performance metrics

Most evaluation metrics for predictive models, when used for classification, are based on the confusion matrix of predicted and actual class labels. To

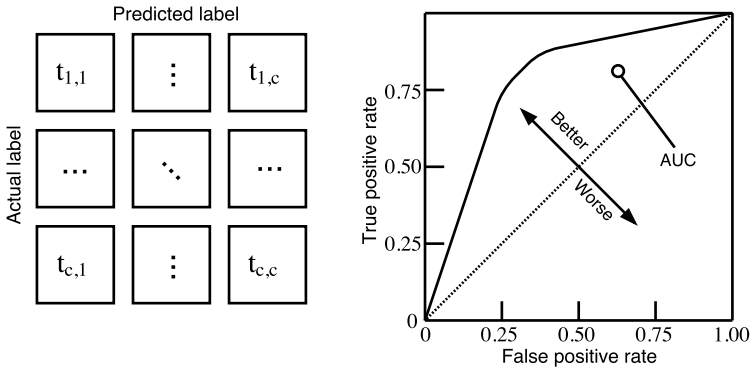

Figure 3.2: Confusion matrix (left) and an example of a ROC curve (right).

illustrate, Figure 3.2 (left) presents the confusion matrix for a classification problem with c classes[1] and T objects, with $T = \sum_{i=1}^{c} \sum_{j=1}^{c} t_{i,j}$ and where $t_{i,j}$ represents the number of examples predicted as $\mathbf{c}_j$ with the actual class $\mathbf{c}_i$. The most common evaluation metric for a classifier is the *accuracy* (or conversely *error rate*), which is simply the fraction of correctly (or incorrectly) classified objects. Using the confusion matrix, the accuracy is defined as

$$A = \frac{\sum_{i=1}^{c} \sum_{j=i}^{c} t_{i,j}}{T}. \tag{3.1}$$

Although accuracy is a useful evaluation metric, it requires the prior class distribution to be balanced, i.e., the categories should be of similar sizes for it to be meaningful. However, in cases where the prior class distribution is unbalanced, e.g., when predicting the occurrence of rare phenomena such as adverse drug events, accuracy may promote models which simply predict the majority class and thus completely ignore the minority class. For imbalanced classification problems, alternative metrics are defined using the confusion matrix. For example, precision, recall (sensitivity), and their harmonic mean (the F-score). Precision is the fraction of correctly labeled predictions of a particular class k:

$$P_k = \frac{t_{k,k}}{\sum_{i=1}^{c} t_{i,k}}. \tag{3.2}$$

Recall or sensitivity, conversely, is the fraction of labels of a particular class k that are correctly predicted as such:

[1]For two class problems, the upper left region is known as the true positives (TP), the upper right region as the false negatives (FN), the lower left as the false positives (FP) and the lower right as the true negatives (TN).

$$R_k = \frac{t_{k,k}}{\sum_{j=1}^{c} t_{k,j}}. \tag{3.3}$$

Similar to sensitivity, specificity is the fraction of labels *not* belonging to a particular class k that are correctly predicted as such:

$$S_k = \frac{T - t_{k,k}}{T - \sum_{i=1}^{c} t_{i,k}}. \tag{3.4}$$

Since there is a trade-off between precision and recall, certain applications might optimize either metric. The F-score incorporates both metrics, and can be weighted to favor either or to provide the harmonic mean (with $\beta = 1$[1]). The F-score of a particular class k is defined as

$$F_k = \frac{(1+\beta^2)P_k R_k}{\beta^2 P_k + R_k} \tag{3.5}$$

To summarize these evaluation metrics, precision, recall, and F-score can be either micro- or macro-averaged. In micro-averaging, binary confusion matrices are constructed, and the sum of the counts to obtain cumulative true positives, true negatives, false positives, and false negatives are used to compute the evaluation metrics. Since micro-averaging tends to favor the majority class [Sokolova and Lapalme, 2009], the metrics employed in this thesis are macro-averaged. In macro-averaging, the mean of a measurement for a particular class k is averaged as

$$\mu_k = \frac{1}{c} \sum_{k=1}^{c} M_k, \tag{3.6}$$

where M is either P, R, S, or F depending on employed evaluation metric.

Since the random forest algorithms investigated in this thesis can output conditional class probabilities, the area under receiver operator curve (AUC) [Bradley, 1997] is employed as an alternative measure of effectiveness. The AUC can, without regard to the prior class distribution or error cost, provide a measure of the trade-off between the rate of true positives and that of false positives. In essence, the AUC measures a predictive model's ability to rank a true positive instance correctly ahead of any false positive instance. A model that can perfectly rank all instances receives an AUC of 1 and a model that provides predictions at random has an expected AUC of 0.5 (see Figure 3.2 (right) for an example of an ROC curve). Although the AUC is mainly employed for binary classification tasks, several extensions have been proposed for multi-class problems [Fawcett, 2006]. In this thesis, a single value for the

[1]The F-score is known as the F1-score if $\beta = 1$.

multi-class AUC is calculated as the mean of the area under the curve for each reference class, weighted by the prevalence of the reference class in the data, as suggested by Provost and Domingos [2000].

The correctness of the conditional probability estimates given by predictive models may, depending on the task, be as important as the ranking performance (as measured by the AUC) or the classification accuracy. One common measure for evaluating the correctness of class probabilities, which is also used in several of the included studies, is the mean square error of probabilities, often referred to as the Brier score. Given class labels $\{\mathbf{c}_1, \ldots, \mathbf{c}_c\}$, let the true class vector of an example $(\mathbf{x}, \mathbf{y})$ be denoted by $\bar{v} = \langle v_1, \ldots, v_c \rangle$, where $v_j = \mathbf{1}(\mathbf{y} = \mathbf{c}_j)$. Also, let $\bar{b} = \langle b_1, \ldots, b_c \rangle$ denote the vector of predicted class probability estimates. Then, the Brier score is defined as $\left\|\bar{b} - \bar{v}\right\|^2$.

The mean square error of class probabilities can be decomposed into bias and variance. Concretely, given the class labels $\{\mathbf{c}_1, \ldots, \mathbf{c}_c\}$, let $\bar{b}_i = \langle b_{i1}, \ldots, b_{ic} \rangle$ denote the vector of probabilities that are predicted by the ith member in the ensemble $\mathbf{H} = \{h_1, \ldots, h_p\}$ for an example $(\mathbf{x}, \mathbf{y})$. Furthermore, let $\bar{v} = \langle v_1, \ldots, v_c \rangle$ denote the true class vector of the example, where, again, $v_j = \mathbf{1}(\mathbf{y} = \mathbf{c}_j)$. Similar to the mean square error (MSE) of a regressor [Geman et al., 1992], the mean square error of class probability estimates of an ensemble is decomposed into bias and variance [Boström, 2012], as follows:

$$MSE = \frac{1}{p} \sum_{i=1}^{p} \left\|\bar{b}_i - \bar{v}\right\|^2 = \left\|\bar{b}_\mu - \bar{v}\right\|^2 + \frac{1}{p} \sum_{i=1}^{p} \left\|\bar{b}_i - \bar{b}_\mu\right\|^2 \tag{3.7}$$

where b_μ denotes the the mean class probability vector. This decomposition allows for investigating whether an observed difference in the mean square error of probabilities between two ensembles is mainly due to the bias, variance or both.

In Paper **VIII**, which concerns early prediction, the average prediction time is employed as a measure of earliness (Equation (2.8)), which measures the average number of time-steps required before a prediction is allowed. Hence, the main interest is to identify a learning algorithm and a trigger function that, in combination, are both accurate and early or provide a reasonable trade-off between the two.

3.2.2 Evaluating classifier performance

As introduced in Section 2.1, the resubstitution error, i.e., the empirical estimate of predictive performance using the same data for both training and testing, of a predictive model, is generally a poor estimate of the expected performance (i.e., how well the induced predictive model approximates the true function that governs the underlying distribution). In particular, since most

machine learning algorithms typically optimize accuracy (using empirical risk minimization), this evaluation procedure results in an overly optimistic estimate of the generalization performance[1]. Since it is intractable to acquire (infinite) additional data drawn from the same underlying distribution, it is often intractable to empirically evaluate the generalization performance. In other words, the learning set constitutes the only data available for learning a model, optimizing its hyper-parameters, and estimating its generalization performance [see Duin, 1996; Salzberg, 1997]. In a typical setting, the learning set is thus divided into parts. These data parts are commonly referred to as the *training set*, the *validation* (or development) set[2], and the *test set* [Alpaydin, 2014].

Instead of estimating the generalization performance of a predictive model using the resubstitution error, which results in a biased performance estimate, predictive models must be evaluated using data not seen during training. These estimates can subsequently be used to, e.g., determine the relative performance of learning algorithms. The most common approach for evaluating a predictive model on unseen data is to separate the learning set into two parts, where one part is used for training (where training constitutes all processes involved in the choice of the final learning algorithm, e.g., hyper-parameters optimization, data pre-processing, and model construction), and one part for testing (estimating the predictive performance). This approach is referred to as the *holdout method*. To guarantee that the training and test data are drawn from the same distribution, both samples are effectively drawn at random from the initial training set.

Although the holdout method provides an unbiased estimate of the generalization performance, it reduces the data available for training the model [Alpaydin, 2014]. The induced model is thus influenced by the ratio of the training data to the testing data. On the one hand, if the training set is too small, the variance of the model increases, adversely affecting the model's performance. On the other hand, the variance of the error estimate increases if the training set is too large, which negatively affects the estimate of generalization performance. To address this, Guyon [1997] suggests, through theoretical work, that roughly 70% of the data should be employed for training and the remaining 30% for testing[3]. To reduce the variability of the holdout method, Breiman et al. [1984] introduce repeated-learning testing, in which the training and testing sets are repeatedly sampled from the training set, and the perfor-

[1]Intuitively, consider a 1-nearest neighbor classifier, which will always perform perfectly when evaluated on the training data.

[2]Since the validation set is essentially a part of the training set, it is sometimes omitted from the discussion.

[3]It is also suggested that the size of the test set can be progressively reduced as the size of the learning set increases.

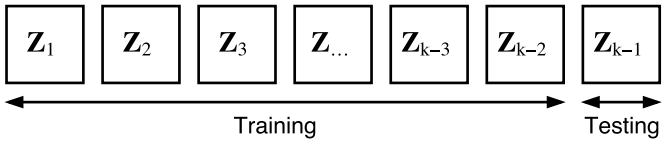

Figure 3.3: Example of k-fold cross-validation

mance is averaged over each round of randomization. Although this approach provides a more reliable estimate of the generalization performance, it does not employ all the training data for learning the model [Dietterich, 1998].

Cross-validation (CV) [Geisser, 1975] has been suggested as an alternative to repeated subsampling, especially when the dataset is small. In brief, CV consists in averaging the performance of several models induced using different splittings of the learning set into disjoint sets of training and testing data. In general, CV consists of partitioning the training set into k disjoint subsets (usually referred to as folds), $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ and, as seen in Figure 3.3, training the learning algorithm using $k-1$ folds, i.e., $\mathbf{Z} \backslash \mathbf{Z}_k$, and evaluating using the remainder, i.e., $\mathbf{Z}_k$. The evaluation over each fold is averaged to provide an estimate of the generalization performance. Thus, at the end of this procedure, each instance has been used exactly once for testing, which means that all test sets are independent. In a typical setting, the number of folds, k, is usually fixed at 10 or 5 [Alpaydin, 2014; Kohavi et al., 1995]; to reduce the variability, it is often suggested to perform cross-validation multiple times, e.g., in configurations such as 10 times 10-fold cross-validation, i.e., cross-validation with 10 folds, repeated 10 times [Arlot et al., 2010]. Irrespective of the validation strategy, stratification is usually employed to ensure that the class distribution is approximately equal for both training and testing.

In this thesis, most of the included studies employ cross-validation to estimate the generalization performance of the investigated algorithms. In Paper **IV**, Paper **VI** and Paper **VIII**, however, already created training and testing splittings (see Section 3.4) are used for ease of comparison between alternative predictive models. It is worth underlining that all reported performance metrics are calculated on test sets which have never been seen by the learning algorithm and that the reported results are in some sense always domain and application dependent.

3.3 Comparing predictive models

As already presented, the methodology adopted in this thesis is conducted quantitatively, using various measures of predictive performance, in an exper-

imental setting, where various factors affecting the induced predictive models are evaluated. The main procedure for providing an answer to the research question thus concerns investigating the relative performance of different predictive models, which amounts to confirming or falsifying hypotheses about their performance. In this context, hypotheses are designed to compare the relative performance, for instance, effectiveness or efficiency, as defined by the chosen performance measure, of several alternative predictive models.

In this thesis, statistical tests are employed to refute the set hypotheses, and, in the case of the null-hypothesis being refuted, identify which methods are responsible for rejecting the hypothesis. In general, the null-hypothesis of no difference is defined as *there is no difference in terms of the selected performance metric between the selected strategies*[1], even if it is not always explicitly stated as such.

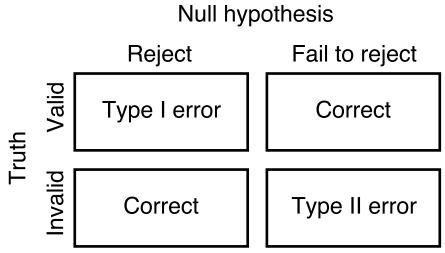

Figure 3.4: The errors of hypothesis testing.

To determine whether differences are significant, a p-value, i.e., the probability of observing an effect given that the null-hypothesis is true, is computed. For the null-hypothesis of no difference to be rejected, the result has to be significant, i.e., the observed p-value must be less than a predefined significance threshold. The significance level consequently determines the probability of rejecting the null-hypothesis when it should be accepted [Alpaydin, 2014]. Figure 3.4 presents the possible outcomes of a hypothesis test. To summarize, a hypothesis test has four outcomes, of which two are desirable, and two are erroneous. More concretely, a Type I error is committed if the null-hypothesis is rejected even though it is valid (i.e., there is no difference) and a Type II error occurs when the null-hypothesis is invalid (i.e., there is a difference), but it is not rejected. To reduce the risk of Type I errors, a low (no greater than 0.05) significance level should be employed. Similarly, to decrease the risk of Type II error, the number of datasets should be sufficiently large. Consequently, the

[1]Strategy refers to any altered factor, be it learning algorithm, hyper-parameters, or other controllable factors.

power of a statistical test is maximized when there is a difference between the predictive models (i.e., the hypothesis is invalid) and the null-hypothesis is rejected, but also correct when there are no differences (i.e., the hypothesis is valid) and the null-hypothesis is not rejected. A significance level of at least 0.05 and a large sample of datasets are employed to reduce the risk of Type I and Type II errors, respectively.

One important distinction when comparing predictive models is whether one is interested in comparing two or more models over one or more datasets. For comparing a pair of predictive models over a single dataset, the most common statistical test is McNemar's test, which expects the agreement between the two algorithms to follow a Chi-square distribution with one degree of freedom [Alpaydin, 2014]. A more common case, however, is to compare the performance of two or more predictive models over two or more datasets. In these cases, the *de facto* standard procedure for evaluating statistically significant differences in terms of a chosen performance metric between predictive models is the non-parametric Friedman test based on ranks [Demšar, 2006; García and Herrera, 2008]. If the Friedman test reveals statistically significant differences, a Nemenyi post-hoc is performed [Demšar, 2006].

3.3.1 Comparing conflicting metrics

In some settings, one is not only interested in evaluating the performance of multiple models over several datasets in relation to a single measure. For example, in Paper **VIII**, which investigates early prediction, the predictive performance of different predictive models is not the only measure of interest. Instead, the model should also (preferably) output the (correct) class label early [see Dachraoui et al., 2015; Xing et al., 2009, 2011]. Although these (in a sense conflicting) metrics provide insights in isolation, there is a trade-off between predictive performance and earliness that should be considered. This trade-off is indeed not captured when the metrics are employed in separation.

One way to reach a conclusion in these situations is to evaluate the two conflicting properties: prediction quality and prediction earliness, in isolation in terms of the Friedman-test over multiple datasets and thereby identify the Pareto optimal model [Dachraoui et al., 2014]. By computing the evaluation metrics in isolation, conclusions can be reached regarding statistically significant differences between the methods under either metric, i.e., prediction quality or earliness. In the framework of Pareto optimality, a predictive model is considered to outperform the alternatives if it improves one objective without significantly degrading the other. Put differently, a predictive model is considered superior to another model if it is significantly better with respect to one of the criteria and *not* significantly worse with respect to the other [Dachraoui

et al., 2014]. In Paper **VIII**, the chose performance metric for prediction quality is accuracy and average prediction time is selected as a measure of earliness.

3.4 Data sources

Benchmark datasets are employed for most empirical work in machine learning, especially when designing and evaluating machine learning algorithms to determine (somewhat) dataset independent differences in predictive and computational performance. In other words, several algorithms are trained, evaluated, and compared over a (large) number of datasets. Although the use of standardized benchmark datasets have been criticized[1] for covering only a limited set of domains (i.e., produce low generalizability), representing small and artificial problems (i.e., have low validity), and promote spurious significant results (i.e., have low reliability) [Salzberg, 1997], they provide several benefits, while the drawbacks can to some extent be remedied by proper statistical tests (statistical tests are discussed in Section 3.3). The main advantage of standard benchmark datasets is that they allow for comparisons between competing approaches to determine, for several domains, significant differences between approaches in a way that promotes repeatable and reproducible research [Salzberg, 1997]. Furthermore, Salzberg [1997] emphasizes that data repositories can be employed to check the plausibility of newly developed algorithms.

For univariate data series, the most widely used standard benchmark datasets are collected in the UCR time series repository [Keogh et al., 2015]. The repository currently contains 85 univariate numerical data series datasets which covers a broad range of domains. These domains are commonly grouped into four categories: *image outline classification*, *motion classification*, *sensor reading classification* and *simulated classification problems* [Lines and Bagnall, 2014]. For multivariate data series, no standard database of datasets exists. Instead, the algorithms included in this thesis for multivariate data series are evaluated using a number of datasets collected from Baydogan and Runger [2014, 2015]. Moreover, to evaluate the predictive performance of random forests induced from heterogeneous data series and discrete data series represented as high-dimensional and sparse matrices, the data has been extracted from the Stockholm EPR Corpus [Dalianis et al., 2012].

In summary, the investigated approaches for handling data series classification, presented in this thesis, are evaluated using different benchmark or real

[1]For example, Salzberg [1997] note that "the evidence is strong that results on the UCI [benchmark] datasets do not apply to all classification problems, and the repository is not an unbiased 'sample' of classification problems".

datasets. In particular, Papers **IV**, **V**, **VI**, and **VIII** are empirically evaluated using univariate and multivariate benchmark datasets from the UCR repository [Keogh et al., 2015], Physiobank [Goldberger et al., 2000], and from Baydogan and Runger [2015]. Papers **I**, **II**, **III** and **VII** are, instead, empirically evaluated using real datasets extracted from the Stockholm EPR Corpus [Dalianis et al., 2012].

Stockholm EPR Corpus

In four of the included studies (Papers **I**, **II**, **III** and **VII**), the investigated approaches are evaluated using data extracted from the Stockholm EPR Corpus [Dalianis et al., 2012], which contains medical records from more than $1,000,000$ patients admitted to one of 512 clinical units within Stockholm County Council during seven consecutive years (2007–2014). In the database, health care episodes are represented by both clinical narratives and structured data regarding, e.g., prescribed drugs[1], diagnoses[2], and clinical measurements. For the earlier studies (Papers I, II and III), a limited subset of the corpus is employed (extracted from the years 2010–2012). For the later studies, however, the full corpus is employed (extracted from the years 2009–2014). A more rigorous description of the employed datasets is provided in the respective studies.

3.4.1 Ethical considerations

Employing medical data for the purpose of constructing predictive models inevitably raises some ethical concerns. These concerns can be viewed from several different perspectives, with the most notable being privacy preservation and scientific transparency. The former is regulated by laws and addressed by the fact that the presented research was approved by the Regional Ethical Review Board in Stockholm (Etikprövningsnämnden i Stockholm), with permission number 2012/834-31/5. Moreover, due to the sensitive nature of the data, datasets constructed from medical records cannot be publicly published, limiting the reproducibility of the presented studies, which impacts the scientific transparency. To address the latter concern, the proposed methods are, in most cases, also evaluated on publicly available benchmark datasets.

[1] Anatomical Therapeutic Chemical classification system (ATC).

[2] International Statistical Classification of Diseases and Related Health Problems—Tenth Revision (ICD-10).

4. Empirical investigations

This chapter summarizes the empirical investigations carried out as part of this thesis. Section 4.1 presents the contributions and main findings of the investigation of how to efficiently and effectively induce random forest from high-dimensional and sparse representations. Section 4.2 presents an empirical investigation regarding a random forest algorithm for directly inducing random forests from (i) univariate, (ii) multivariate and (iii) heterogeneous data series. Next, the random forest introduced for univariate data series is revisited for early prediction of time series. Finally, Section 4.3 presents an empirical investigation of random forests for the purpose of predicting adverse drug events.

4.1 Random forests for classifying high-dimensional data

In this section, two approaches for improving the predictive performance of random forests for high-dimensional and sparse representations, extracted from medical data series, are explored in response to the first objective. In the first part, a random projection technique in conjunction with random forests is adapted to project discrete data series to a low-dimensional space. In the second part, the random forest algorithm is modified to improve predictive performance on high-dimensional and sparse datasets. While these contributions are posited in relation to data series classification, the results from the investigation generalize to any high-dimensional or sparse data.

4.1.1 Background

Although machine learning techniques, including random forests, are, with varying degrees of success, able to cope with large, high-dimensional and sparse data, the predictive and computational performance tends to deteriorate on such representations [Goldstein et al., 2011]. One approach to overcome this is to use attribute selection algorithms or dimensionality reduction techniques, as discussed in Section 2.2. An alternative, however, is to address the issue in the learning algorithm. In the literature, several approaches have been proposed for various learning algorithms, e.g., logistic regression [Shevade and Keerthi, 2003]. However, limited attention has been given to random forests

and decision trees, especially when the data is highly sparse.

For random forests, the problems introduced by sparsity are the risk of selecting a subset of uninformative attributes at each splittin, which results in shallow decision trees with high bias. By selecting uninformative attributes, the predictive performance of the induced models is thus negatively affected, which can be explained by the fact that the majority of the trees in the forest simply predict the majority class. To avoid ending the tree construction prematurely, the most common approach is to simply increase the number of investigated attributes at each splitting [Goldstein et al., 2010], which decreases the risk of selecting uninformative attributes. Although increasing the number of attributes indeed allows for improving the predictive performance, an arbitrarily large sample of attributes is required, introducing the need and cost of manual parameter tuning. Furthermore, by increasing the number of inspected attributes at each node, the computational efficiency and variability of the trees is reduced[1]. There is thus a trade-off between the number of attributes to investigate, the computational efficiency, the variability of the forest, and the bias of the trees.

In summary, there are many challenges when inducing predictive models from high-dimensional and sparse data. Consequently, in response to the first objective, Paper **II** and Paper **III** aim to investigate ways of mitigating some of these problems for random forests. In the first, study the problem is addressed by employing a, for the context novel, dimensionality reduction technique, and in the latter case, by modifying the splitting criterion in the random forest algorithm.

4.1.2 High-dimensional random forests

In Paper **III**, two resampling based approaches for reducing the effects of data sparsity and high dimensionality are explored and compared to two baseline approaches. The re-sampling based approaches, introduced in this thesis, rely on the extending the splitting procedure to explore additional attributes in cases where no informative attributes (information gain equals 0) are found in the initial sample[2]. Paper **III** explores the following approaches ($|\mathbf{A}|$ denotes the number of attributes in the dataset):

a) Sampling $l = \lfloor \log_2 |\mathbf{A}| + 1 \rfloor$ as suggested by Breiman [2001] (denoted by `log`).

[1]In the extreme where all attributes are investigated, randomization is only introduced in the training data.

[2]In Paper **III**, the initial sample consists of $l = \lfloor \log_2 |\mathbf{A}| + 1 \rfloor$ attributes

b) Sampling $l = \lfloor \sqrt{|\mathbf{A}|} + 1 \rfloor$ as suggested by Díaz-Uriarte and De Andres [2006] (denoted by `sqrt`).

c) Re-sampling l new attributes at least d times, if no informative attributes are found (denoted as `resample`).

d) Re-sampling a single new attribute until an *informative* attribute is found (denoted by `retry`).

Although the re-sampling based splitting conditions (i.e., c and d) require the decision tree algorithm to be slightly altered, by injecting more randomness into the tree generation process the trees are expected be more diverse while reducing the individual tree's bias (i.e., the tendency for predicting the majority class). Specifically, given that an informative attribute exists, the `retry` approach is guaranteed to find an informative attribute. Similarly, `resample` can be seen as a cost controlled way of reaching `retry` via an adjustable hyper-parameter, namely the number of re-samples to investigate.

4.1.3 Random indexing for discrete data series

Random indexing is a technique, similar to random projection [Bingham and Mannila, 2001], which has historically been used for reducing the dimensionality of high-dimensional documents, in order to preserve word meaning in distributional word-space models [Kanerva et al., 2000]. In random indexing, each word is assigned an empty (all zero) context vector and a sparse (index) vector consisting of a large number of zeros and a small number[1] of randomly distributed 1s and -1s [Sahlgren, 2005]. Both the context vector and the index vector have the same dimensionality, which is chosen to be significantly smaller than the language's vocabulary. In this model, each word's context vector is updated by adding up its surrounding words' index vectors, resulting in an approximation of the term similarites.

In Paper **II**, a novel use of the random indexing algorithm is suggested to reduce the dimensionality of discrete event series by ignoring the order dimension. Given a dataset $\mathbf{X}$ consisting of n discrete univariate data series of m events, where each data series, $\mathcal{T} \in \mathbf{X}$, consists of events $T^1, \ldots, T^m$, assign every unique T^j a random index vector of length d denoted as r_j. By this construction, each data series $\mathcal{T}$ can be represented by the vector

$$\mathbf{x} = \sum_{j=1}^{m} \hat{r}(T^j), \tag{4.1}$$

[1] Between $1-2\%$ of the elements are usually either 1 or -1.

where $\hat{r}(T)$ returns the index vector of event T. By employing this algorithm, the dimensionality of a binary vector representation constructed from a discrete data series can be reduced to a size determined by d, which is a tuning parameter introduced when generating the index vectors, r_j. Specifically, since the index vectors are *nearly* orthogonal [Kanerva et al., 2000], the instances can be represented as dense numerical vectors instead of sparse binary item-counting vectors. The idea is that this process significantly compresses the original data, and by doing so improves the computational and (hopefully) predictive performance when applying a learning algorithm.

In Paper **II**, three different settings for the random indexing algorithm are considered. These settings depend on the number of bits (1 and -1) used to represent the random index vector and the size of the resulting context and index vectors. Hence, to investigate the effect of dimensionality reduction on the chosen performance measures, the explored settings consider varying degrees of dimensionality reduction which in the included study amounts to 5 (`small`), 10 (`medium`), and 20 (`large`) percent of the original dimensionality and includes $\approx 0.5\%$ non-zero bits per vector. These settings were chosen to minimize the risk of conflicts, i.e., the risk of two items from the data series sharing the same index vector, while at the same time not reduce the computational efficiency.

4.1.4 Study design

The empirical investigations carried out to detecte whether there are any significant differences in efficiency and effectiveness between the different predictive models induced from high-dimensional and sparse data were done in the context of detecting adverse drug events from patient data extracted from an electronic health record system. The data source used in these experiments consisted of discrete data series, extracted from the Stockolm EPR Corpus [Dalianis et al., 2012], of drugs and diagnoses prescribed and assigned to patients up to the point when they were diagnosed with a particular ADE. Both Paper **II** and Paper **III** use as positive examples patients which has been diagnosed with a particular ADE related diagnosis code and other, randomly selected, patients without the specified code as negative examples. The discrete data series were transformed to high-dimensional and sparse representations by indicating the presence or absence of the prescribed drugs or assigned diagnoses.

4.1.5 Main findings

In Paper **III** it is shown that there is a significant difference in predictive performance as measured by the AUC and the F-score between the investigated

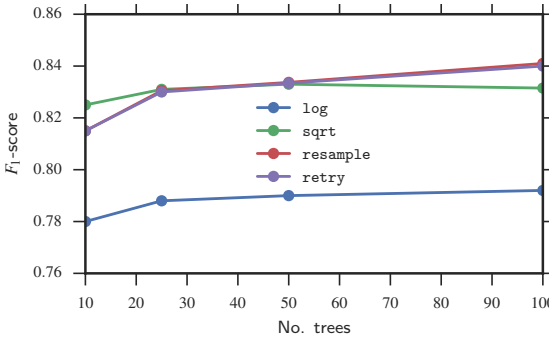

Figure 4.1: Average F-Score for different splitting conditions and forest sizes. This figure is adapted from Paper **III** with permission.

approaches, i.e., between sampling-based (c and d) and non sampling-based (a and b) approaches. The empirical investigation shows that the differences are most prominent for smaller forest sizes, in which case the baseline approach `sqrt` performs significantly better than the alternatives (especially for F-score, see Figure 4.1). Interestingly, however, if more trees are added to the forest (which would be the normal thing to do if aiming for maximum predictive performance), the re-sampling based approaches tend to outperform the baseline, with the best performing approach being `retry`.

As shown in Paper **III**, the difference in predictive performance can, using the bias–variance trade-off (see Section 2.5.1), to some extent be explained by the fact that while increasing the number of attributes results in decision trees that are more correct on average (lower bias), the re-sampling approaches result in higher variability (see Figure 4.2) in the individual predictions made by the trees. Since the variability of the individual models can be averaged out, this seems to lead to improved performance of the forest, provided that it contains sufficiently many trees.

The study presented in Paper **III** demonstrates that the choice of approach for handling high-dimensional and sparse data in random forests is highly dependent on the performance measure which is most relevant for the application domain. The study highlights the importance of the sampling-based approaches, especially `resample`, when the goal is to produce as accurate a model as possible. On the contrary, however, the empirical investigation indicates that the baseline approach that simply increases the number of explored attributes at each node outperforms the alternatives when the aim is to assign the probability of a certain class accurately. Finally, the empirical investigation reveals that the choice of approach for handling high-dimensional sparse data is of minor importance if the goal is to maximize ranking performance.

In Paper **II**, the feasibility of employing random indexing for reducing the

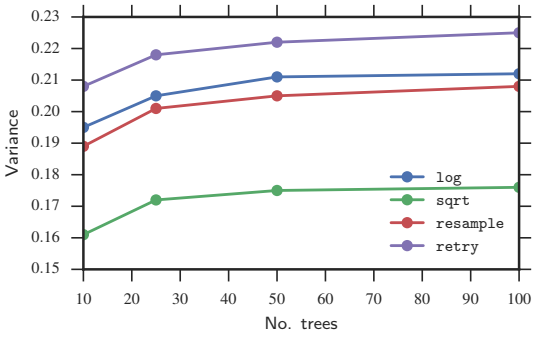

Figure 4.2: Average variability for different splitting conditions and forest sizes. This figure is adapted from Paper **III** with permission.

Table 4.1: The impact of dimensionality reduction on the performance measures. Statistically significant deviations in favor of reducing dimensionality are denoted by †. Deviations in favor of not reducing dimensionality are denoted by *. The table is reproduced from Paper **II** with permission.

	Sensitivity	Specificity	AUC	Relative training time
`small`	**0.604**†	0.916	0.9217	**0.125**†
`medium`	0.598†	0.914	0.9210	0.143†
`large`	0.584†	0,935	**0.9220**	0.155†
baseline	0.466	**0.982***	0,9100	1

dimensionality over item-counting methods is investigated. The main finding in Paper **II** is that the efficiency of inducing predictive models using the proposed method is greatly improved (including the cost of dimensionality reduction), without sacrificing predictive performance. In fact, the empirical investigation indicates that the AUC of predictive models induced from instances with a reduced dimensionality, on average, improves slightly over a baseline random forest employing the `retry` approach from Paper **III** (these results are summarized in Table 4.1).

4.2 Random forests for data series classification

In this section, random forests for data series classification are empirically investigated. In direct response to the second objective, the exploration evolves from univariate (Paper **IV**) and multivariate (Paper **V** and **VI**) data series classification to heterogeneous data series classification (Paper **VII**). Finally, in response to the third objective, the main findings from an empirical investigation

of random forests for early time series prediction (Paper **VIII**) are presented.

4.2.1 Background

As introduced, data series are common in many machine learning applications, where measurements of some quantity are made over time or over other logical orderings. Achieving fast, accurate and early prediction of data series has, as a result, attracted significant interest in the machine learning community. Although these representations can be transformed into traditional multidimensional data, e.g., as presented in Section 4.1, there are many problems with such representations. For univariate data series, experimental evidence has repeatedly demonstrated [Bagnall et al., 2016; Wang et al., 2013; Xi et al., 2006] that prototype based methods (see Section 2.3.1) with elastic distance measures, such as dynamic time warping, provide some of the most accurate predictive models in this domain. These models are however both prone to overfitting and, while quick to train, in the case of large datasets are slow to use for predictions. Current research thus increasingly focuses on parametric or non-parametric phase-dependent, phase-independent, or dictionary-based features, since they are faster to apply for prediction and can in many cases improve the generalization performance [Bagnall et al., 2016]. Single shapelet-based decision trees (see Section 2.4.3) have been proposed as a viable alternative, but these models are slow to train and have limited predictive performance. Hence, a natural extension is to consider random forests of such trees.

Furthermore, in a growing collection of applications, instances are not represented by single dimension but by several dimensions. As mentioned, multivariate data series are characterized not only by similarities between individual dimensions but also by relations between different dimensions. Since the latter is not captured by traditional univariate strategies [Bankó and Abonyi, 2012], the attention for multivariate data series has been directed towards feature-based strategies (see Section 2.3.2), such as learned pattern similarity [Baydogan and Runger, 2015], that are better suited to capturing these similarities. In the context of multivariate data series, a natural extension of the shapelet-based decision tree is to consider multivariate splits and, consequently, random forests of such trees. Similarly, shapelets, i.e., short discriminatory subsequences, are by nature well-suited for early classification, since the full data series is not required at prediction time.

Therefore, in response to the second objective, various ways are evaluated for constructing random forests from (i) univariate, (ii) multivariate and (iii) heterogeneous data series in order to investigate their efficiency and effectiveness. Similarly, in response to the third objective, this thesis investigates the effectiveness of the introduced random forest for the purpose of performing

early prediction of data series.

4.2.2 Random shapelet forest

Paper **IV** introduces a random forest based on randomized shapelet trees (denoted by `RSF`). The proposed algorithm revolves around the concept of shapelet-based decision trees (introduced in Section 2.4.3) for which a (limited) set of random shapelets is employed during training. By employing a limited set of (random) shapelets, the efficiency of the tree induction algorithm can be significantly improved over the standard shapelet tree. Moreover, by employing ensemble model averaging, the variance introduced by randomization can be averaged out with the idea that this, similar to traditional random forests, improves the predictive performance. The algorithm for finding the best shapelet splitting (Algorithm 4), is thus modified to, instead of extracting all shapelets on Line 3, extract a subset of random shapelets. Apart from the natural hyper-parameter of the number of ensemble members, the random shapelet forest has three hyper-parameters that require tuning for different applications. These parameters are the number of random shapelets to extract, the minimum shapelet length, and the maximum shapelet length. The impact of these parameters is studied in Paper **VI** through an extensive empirical investigation.

In Paper **V**, the random shapelet forest is extended to handle multivariate data series, by mapping the problem to a split-and-combine framework. In the splitting step, a dataset $\mathbf{X}$ of d-dimensional data series is split into d datasets $\mathbf{X}^d$ of univariate data series. From these datasets, a single predictive model is induced from each individual datasets by $\mathbf{H} = h_1, \ldots, h_d$. Given a new previously unlabeled multivariate data series $\mathcal{T}$, it is split into d univariate data series $\mathbf{T}_1, \ldots, \mathbf{T}_d$ and each base-model in h_k is applied to the kth dimension, $\mathbf{T}_k$, and combined using a suitable ensemble approach. In Paper **V**, the following combination approaches are empirically evaluated:

a) Majority voting (see Equation (2.16))

b) Distributional summation (see Equation (2.17))

c) Performance weighting

Since the training set limits the variability of the ensemble using the approach presented in Paper **V**, Paper **VI** generalizes the random shapelet forest to generate multivariate shapelet-based decision trees. In these trees, both the selection of the shapelet and the dimension from which it is extracted is randomized. To support heterogeneous data series, Paper **VII** further generalizes the random shapelet forest to consider any kind of sequential pattern. In Paper **VII**, heterogeneous data series of both discrete and numerical values are

examined, in the context of three different distance measures and three pattern extraction procedures. For discrete data series, the edit distance and zero–one distance are employed, and for numerical data series, the Euclidean distance is employed. A complete explanation of all algorithms is presented in the included studies.

4.2.3 Early random shapelet forest

One approach to enabling early prediction using the random shapelet forest is to construct one forest per time-step and assess the earliest time where a classification can be made using cross-validation, similar to the approach presented by Mori et al. [2016]. This approach is, however, computationally costly and requires a large number of models to be built and evaluated. To overcome this limitation, the main idea in Paper **VIII** is that since a significant portion of the randomly sampled shapelets of each tree will be shorter than the incomplete data series, only the tree prediction (traversal) function require modification.

Paper **VIII** investigates two approaches to allowing the random shapelet forest to support early prediction. The proposed approaches are motivated from how missing values are handled in traditional decision trees using either weight propagation or class proximities while traversing the trees. In short, the former employs an algorithm similar to how missing values are handled in traditional decision trees [Quinlan, 1986], where the prediction made by a leaf is weighted by the probability of reaching the node in the case of missing data (denoted as `WERSF`); and the latter employs a procedure similar to how surrogate splittings are handled in traditional decision trees (denoted as `PERSF`). Finally, to determine the earliest time instant when a final prediction is allowed, the reliability threshold proposed by Mori et al. [2016] is extended and estimated using the out-of-bag instances (see Section 2.5.2). Specifically, the earliest time point for each class is estimated such that the all later time instants have a precision that is higher than the precision of the full model, weighted by a user chosen constant (w). Hence, the weighting constant is used to trade timeliness for predictive performance or vice versa.

4.2.4 Study design

The empirical investigation carried out to investigate random forests for data series classification were done in the context of benchmark datasets. In Paper **IV**, the random shapelet forest for univariate data series classification is compared to three different nearest neighbor approaches using different distance measures over 45 datasets from the UCR time series repository [Keogh et al., 2015]. This evaluation was extended in Paper **VI** to include a large selection of state-of-the-art methods for univariate data series classification

Table 4.2: Summary of the strategies to which the investigated random forest (denoted as RSF) are compared.

Predictive model	Publication	Abbreviation
Nearest neighbor (Euclidean distance)	e.g., Ding et al. [2008]	NN
Nearest neighbor (cDTW)	e.g., Wang et al. [2013]	cDTW
Nearest neighbor (cDTW-cv)	e.g., Wang et al. [2013]	cDTW-CV
Shapelet tree	Ye and Keogh [2009]	FST
Shapelet transform	Hills et al. [2014]	ST
Fast shapelet trees	Rakthanmanon and Keogh [2013]	SAX
Learning shapelets	Grabocka et al. [2014]	LTS
Ultra-fast shapelet transform	Wistuba et al. [2015]	UFS
Learned pattern similarity	Baydogan and Runger [2014]	LPS
Symbolic representations of time series	Baydogan and Runger [2015]	SMTS
Early classification of time series	Xing et al. [2009]	ECTS
Reliable early classification	Mori et al. [2016]	ECDIRE

and multivariate data series classification. In Paper **VII**, heterogeneous data series representing prescribed drugs, assigned diagnoses, clinical and laboratory measurements were extracted from the Stockholm EPR Corpus for the purpose of predicting ADEs. The effectiveness of random forest for heterogeneous data series classification was in this context compared to random forests (using the `retry` approach) induced from the high-dimensional representation described earlier. Finally, the early random shapelet forest was compared to several state-of-the-art methods for early classification over 45 benchmark datasets. Table 4.2 presents an overview of the algorithms compared to in the empirical investigation.

4.2.5 Main findings

In Paper **IV**, a randomization-based ensemble strategy, the random shapelet forest, for classifying data series, based on random shapelets, was introduced and empirically evaluated. The empirical results show that the introduced algorithm significantly outperforms prototype-based predictive models under var-

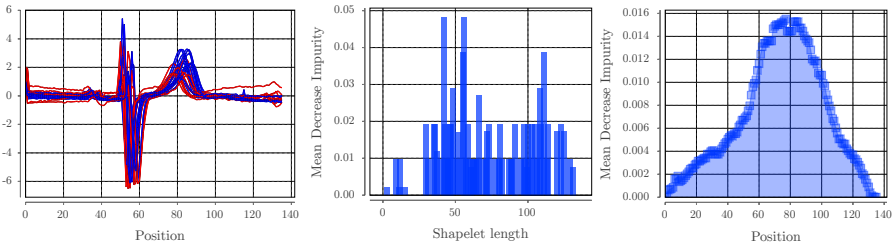

Figure 4.3: The time series in the `ECGFiveDays` dataset with the respective length and positional importance scores. This figure is reproduced from Paper **IV** with permission. Best viewed in color.

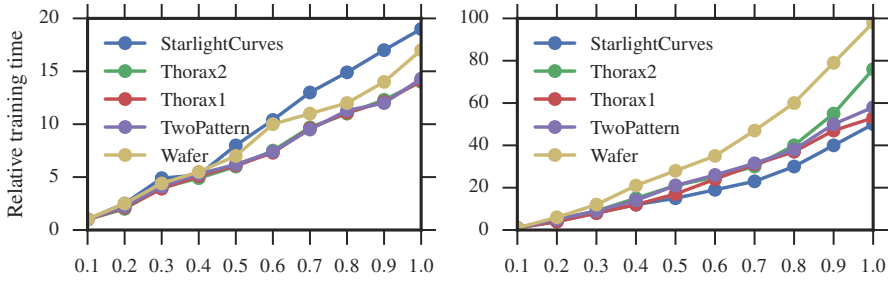

Figure 4.4: The relative training time of the random shapelet forest as a function of increasing training set size (n) and data series length (m). This figure is reproduced from Paper **VI** with permission. The paper also also provides information about the datasets. Best viewed in color.

ious distance measures for the task of classifying univariate numerical data series in terms of predictive performance, as measured by accuracy. It was also shown that by adapting the impurity decrease measure, important regions of time series could be extracted and employed for analyzing and interpreting the resulting forests. An example of the resulting importance score can be found in Figure 4.3, which shows both the important regions and the important shapelet sizes.

In Paper **V**, several strategies for combining predictive models individually induced over multiple dimensions are evaluated. Several different strategies were evaluated and compared using different base models, consisting of prototype-based methods and random shapelet forests. The empirical evaluation shows that combining random shapelet forest models significantly outperforms the alternatives. The experiment also highlights that the best performing combination strategy is weighted majority voting (c), for both base-models.

In the study presented in Paper **VI**, the random shapelet forest is generalized to directly induce predictive models from multivariate data series and to evaluate the impact of the hyper-parameters. The study specifically demon-

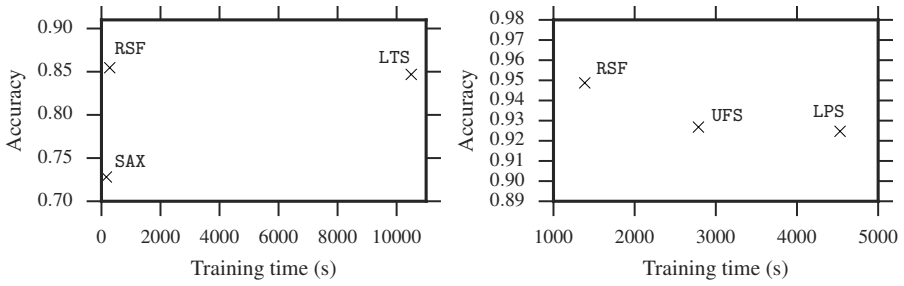

Figure 4.5: The trade-off between effectiveness as measured by accuracy and efficiency as measured by training time (in seconds). The right figure shows `RSF` compared to `UFS` and `LPS` for multivariate numerical data series. The left figure shows `RSF` compared to `SAX` and `LPS`. The optimum lies to the top left of the figures.

strates that by combining several relatively weak random shapelet trees in an ensemble, substantial gains in predictive and computational performance can be achieved, compared to both single shapelet trees and a wide array of other learning algorithms. For univariate (numerical) data series, it is demonstrated through an extensive empirical investigation that the proposed algorithm yields predictive performance comparable to the current state-of-the-art (`LTS` and `ST`) and significantly outperforms several alternative algorithms (`cDTW-CV`, `FST`, and `SAX`) while being at least an order of magnitude faster. Similarly, for homogeneous (numerical) multivariate data series, it is shown that the random shapelet forest utilizes significantly less computational resources while being more accurate than the current state-of-the-art, i.e., when compared to `UFS`, `LPS`, and `SMTS`. Follow-up investigations were carried out, which investigate the effect of hyper-parameters on the bias–variance decomposition. These investigations reveal that the optimal number of shapelets is indeed related to the bias–variance trade-off, where investigating more shapelets gives trees that are more accurate on average but are also more correlated, and investigating fewer shapelets at each node, which gives weaker but less correlated trees.

Concerning efficiency, the empirical investigation confirms the theoretical run-time and indicates that the presented algorithm scales approximately linearly in the size of the training set and quadratically in the data series length (see Figure 4.4). In terms of the trade-off between effectiveness and efficiency, Figure 4.5 shows the accuracy and the training time for the `RSF` algorithm compared to two alternatives for homogeneous multivariate data series (right) and for univariate data series (left). The random forest introduced in Paper **VI** is shown to provide a good trade-off between training time and accuracy compared to these alternatives. Finally, although not included in any of the presented studies, the generalized random shapelet forest (Paper **VI**) is empir-

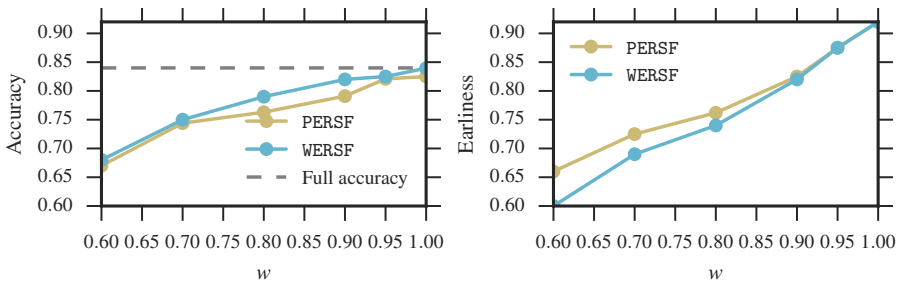

Figure 4.6: The predictive performance and earliness for different multipliers of the full precision. This figure is adapted from Paper **VIII** with permission.

ically shown to outperform the combination strategies evaluated in Paper **V** significantly[1].

In Paper **VIII**, the early random shapelet forest is presented, and two different strategies for enabling early prediction are empirically evaluated. The investigation reveals that the introduced algorithm and evaluated strategies significantly outperform the competing methods in terms of predictive performance. Specifically, the presented study shows that the weight propagation strategy performs better regarding both predictive performance and earliness compared to the proximity based method. The study also reveals that the early random shapelet forest, irrespective of the strategy employed, performs unfavorably in terms of earliness compared to the alternatives. Regarding both evaluation metrics, the empirical investigation reveals that the performance of the introduced algorithm cannot be significantly separated from the alternatives, i.e., `ECTS` and `ECDIRE`. To investigate the effect of the weight constant w, the earliness and accuracy for the proposed methods are computed using cross-validation on the training set. In Figure4.6, the evolution of early classification accuracy (left) and earliness (right), for both approaches. As seen, the earliness and accuracy increases, as expected, with an increasing value of w. Specifically, when setting the w parameter to be maximally conservative ($w = 1$), the accuracy approaches the accuracy of the complete model for both methods. Furthermore, the precision threshold seems to, on average, empirically lower bound the accuracy of the forest on the full time series, indicating that the threshold indeed functions as expected and provide means for trading accuracy for earliness or vice versa.

[1]For the task of detecting heart related problems from ECGs extracted from Physiobank (see Paper **V** for a complete explanation of the data), the best combination strategy in Paper **V** achieves an accuracy of roughly 85% compared to the generalized random shapelet forest (Paper **VI**) which obtains an accuracy of approximately 92%.

4.3 Adverse drug event detection

In this section, random forests are considered as predictive models for the detection of adverse drug events, explored from three perspectives. First, in Paper **I**, the feasibility of predicting ADEs from electronic health record data is demonstrated. Second, in Paper **II** and Paper **III**, the predictive and computational performance of predicting ADEs from high-dimensional and sparse data is investigated. Finally, in Paper **VII**, the effectiveness of random forests adapted for data series classification is evaluated in the context of ADE detection. Although this empirical investigation is not directly related to the research question and objectives of the dissertation, in this way the investigated algorithms are evaluated in relation to the practical motivation for this thesis.

4.3.1 Background

Adverse drug events, defined as undesired harms caused by the intake of medications [Nebeker et al., 2004], account for an increasing amount of hospitalizations and deaths worldwide [Beijer and De Blaey, 2002; Howard et al., 2007]. Adverse events are thus both a serious health concern, estimated to be the seventh most common cause of death in Sweden [Wester et al., 2008], and a significant burden on the health care system [Schneeweiss et al., 2002]. Although a benefit–risk analysis of newly developed drugs is already conducted during clinical trials, post-marketing detection and surveillance are necessary to detect unanticipated events. For instance, clinical trials are normally performed with a limited sample of patients, who are followed for a limited period of time. As a result, not all serious adverse events can be detected prior to market deployment, which results in drugs being withdrawn from the market due to serious adverse reactions not detected during clinical trials. For example, Cerivastatin, a drug used to lower cholesterol and prevent cardiovascular diseases, was withdrawn worldwide in 2001 since it could cause fatal rhabdomyolysis [Furberg and Pitt, 2001]. Similarly, Valdecoxib was withdrawn from the market in 2005 because of serious dermatological conditions, not detected during clinical trials [Zhang et al., 2006].

The activities related to the detection, signaling, and assessment of adverse drug events is referred to as pharmacovigilance or post-marketing drug surveillance. During post-marketing surveillance, a vast array of automatic approaches for detecting potential safety hazards of drugs have been investigated, [e.g. Almenoff et al., 2007; Pariente et al., 2007], using various data sources, the most prominent of which is a disproportionality analysis of spontaneous individual case reports [Suzuki et al., 2010]. One of the main obstacles, however, with current systems for collecting and analyzing data regarding adverse

drug events is the fact that serious ADEs are heavily under-reported, while known ADEs are over-reported, by both clinicians, in the case of EHRs, and by patients, in the case of individual case reports [Hazell and Shakir, 2006]. Complementary and alternative sources have thus been investigated, such as online communities [Liu and Chen, 2013] and EHRs. The major benefit of EHRs is that they typically contain longitudinal observational data of large samples of patients, including demographic information, medical history, drug consumption with exposure time and dose information, and clinical measurements, including lab results and drug concentrations [Dalianis et al., 2012]. To improve the reporting rate, systems have thus been investigated to automatically detect ADEs from electronic health records, which avoids several of the limitations present in case reports [Zhao et al., 2014, 2015]. The design of such systems, however, presents, as already introduced, many challenges.

4.3.2 Main findings

In Paper **I**, **II**, **III**, and **VII**, the task of predicting adverse drug events from health care episodes is tackled from various perspectives. Paper **I** demonstrates that it is possible to predict adverse drug events by simply representing health care episodes as binary vectors indicating the presence or absence of medications and diagnose. In Paper **II** and Paper **III**, large-scale empirical investigations confirm the possibility of predicting adverse drug events from health care episodes and that by reducing dimensionality and handling sparsity, the predictive performance can be significantly improved. In Paper **VII** it is further demonstrated that by taking temporality into consideration and thereby enabling the introduction of laboratory measurements and drug concentrations, the predictive performance of the task can be significantly improved.

5. Concluding remarks

In this chapter, a wider discussion of the main focus of this thesis will be revisited. The three outlined objectives will be revisited to provide an answer to the research question of this thesis. Finally, this dissertation will end by presenting future directions.

5.1 Discussion

This dissertation set out to investigate how efficient and effective random forest models can (a) be induced from high-dimensional and sparse data, (b) be induced from univariate, multivariate and heterogeneous data series, and (c) be induced so as to render predictions early. These objectives were formalized in an overall research question which entailed the investigation of effective and efficient induction for random forests to support the classification of data series.

5.1.1 Random forests for classifying high-dimensional data

As pointed out in the Introduction, representing discrete data series using high-dimensional and sparse vector spaces has been shown to be detrimental to the predictive and computational performance of many learning algorithms [Caruana et al., 2008]. For example, as already shown, in cases where the data is both sparse and high-dimensional, the random forest algorithm often constructs models which are severely biased towards predicting the label with the majority *a priori* probability simply because too few attributes are inspected at each node. Similarly, reducing the dimensionality of such representations has been shown to be rather inefficient [Cao et al., 2003]. To address some of these limitations, this thesis investigates approaches for efficiently and effectively handling such representations from different perspectives. These perspectives were to (a) change the way in which the splitting conditions in the random forests are chosen and (b) investigate a dimensionality reduction technique that is novel for this task.

Both these techniques were evaluated on health care data extracted from a real EHR system, representing patients who had been diagnosed with an ad-

verse drug event. Ignoring the methodological implications of the investigated strategies, this allowed the empirical investigation not only to highlight differences in predictive performance but also to demonstrate the feasibility of the task of predicting adverse events (discussed in Section 4.3). In Paper **III**, it was shown in a rather extensive empirical investigation that the choice of approach for handling sparsity in random forests is dependent on the performance metric of interest. Specifically, if the task is to accurately assign an ADE to a patient record, a sampling based approach can be recommended. On the other hand, if the task is to rank patients according to the risk of a certain ADE, the choice of approach was shown to be of minor importance. One of the main advantages of the proposed methods was that, by allowing the random forest algorithm to dynamically adapt the number of employed attributes, the need for hyper-parameter tuning was mitigated, which simplifies the algorithms. It should, however, be noted that this investigation was non-exhaustive and employed a limited set of datasets from a limited domain. As a consequence, it is possible that other approaches might allow for improvements in predictive performance or that other domains highlight other performance characteristics.

In Paper **II**, which is related to the first objective, a pre-processing step was investigated to alleviate the problem of high sparsity, primarily in terms of the efficiency of the model induction. Although the empirical investigation showed the utility of the presented method, both in terms of increased predictive performance and in reduced computational cost, approaches for increasing the variability among the members of the forest could have been investigated. For example, this could have been achieved by employing bootstrapping during the dimensionality reduction for each tree, similar to how rotation forests utilize principal component analysis [Rodriguez et al., 2006]. Moreover, a clear limitation of the study is that there is no comparison to any related dimensionality reduction or feature selection method, which limits the ability to draw conclusions regarding the relative performance of the proposed method. Hence, to improve these experiments, alternative dimensionality reduction techniques should have been included in the investigation to allow more detailed discussions regarding the effectiveness of random indexing in this context. As a result, the study serves as a demonstration of the methods utility, but not its (relative) usefulness.

5.1.2 Random forests for data series classification

Three extensions to the random forest framework for classifying data series was put forward and evaluated in this dissertation. These extensions were introduced in direct response to the second objective, which explored efficient and effective algorithms for directly inducing random forests from data series.

The investigated extensions of the random forest were to handle (i) univariate data series, (ii) homogeneous (numerical) multivariate data series, and (iii) heterogeneous (discrete and numerical) data series.

In Paper **IV**, it was demonstrated that by employing local shapelets in a forest of randomized trees, which was termed the random shapelet forest, significant improvements in terms of predictive performance could be achieved when compared to related algorithms. This limited empirical evidence in favor of the random shapelet forest was, in Paper **VI**, confirmed in a large-scale comparison. In particular, Paper **VI** demonstrated that the random shapelet forest can trivially be generalized to homogeneous (numerical) multivariate data series.

The extension to multivariate data series allowed the forest to model complex dependencies, not only within a single dimension but also between different dimensions, while jointly taking into account the order of values. This formulation, as it turns out, seemed to work very well in relation to other related state-of-the-art algorithms presented in the literature [e.g. Baydogan and Runger, 2015; Grabocka et al., 2014]. Moreover, when comparing the two strategies for multivariate data series introduced in Paper **V** and Paper **VI**, empirical evidence revealed that the latter, which constructs multivariate shapelet-based decision trees, outperformed the former, which constructs a single model per dimension, lending some confirmation to the fact that relations between different dimensions are important when classifying multivariate data series. For future work, it would be important to further understand these relations in more detail, e.g., by taking the order of patterns from different dimensions into account simultaneously.

Further developments of the random shapelet forest were investigated in a follow-up study (Paper **VII**), in which alternative sequential patterns were investigated. This experiment showed that for the purpose of predicting adverse drug events using heterogeneous data series extracted from electronic health records, taking temporal dependencies into consideration improves the detection rate and thereby the predictive performance. Since the exploration of patterns was non-exhaustive, future directions would be to consider alternative pattern representations.

Although not thoroughly discussed in this dissertation, the heterogeneous data series extracted from the EHR system contains missing dimensions since not all patients have had all measurements performed. In particular, consider the patients represented by the value of laboratory measurements. For such representations, not all measures are present for all patients. Missing value imputation is a well-studied area, but it is currently unknown how to impute entire dimensions for data series classification. A clear limitation of Paper **VII** is that the problem of missing values was tackled by simply considering in-

stances lacking the dimension from which the pattern was extracted as maximally different. This approach does not, however, consider the importance of the missing attributes. For future work, surrogate splits or other missing value strategies common to decision trees should be investigated when dealing with data series with missing dimensions.

Modeling data series directly in the learning algorithm has, as shown in this thesis, many advantages. On the one hand, it in part mitigates the problem of learning from sparse and high-dimensional data by considering the temporal dimension and identifying patterns thereof. On the other hand, it allows for constructing predictive models that are able to output labels early, i.e., in a setting where examples are streaming one value at a time. Strategies for learning such early predictive models using random forests were investigated in Paper **VIII**, in response to the third and final objective. This approach was shown to utilize less computational resources than alternative state-of-the-art methods since a single classifier was built that utilized the full sequences and a modified prediction function.

Paper **VIII** showed that, depending on how the weighting parameter was set, accuracy vs. earliness could be balanced in different ways. The empirical evaluation, in which the weight was optimized towards accuracy, showed that the proposed approach, in two different instantiations, could lead to significantly higher predictive performance than the alternatives. On the other hand, for this optimization criterion, the improved accuracy came with an associated cost of reduced earliness. By tuning the parameter differently, earliness was shown to be improved, but this improvement again came at the cost of predictive performance. Since the model does not need to be retrained for different weighting parameters, the presented strategy provides a convenient framework for evaluating various trade-offs between accuracy and earliness at a low computational cost. An important limitation of the early random shapelet forest is the lack of theoretical support for the process in which the earliness threshold is found.

5.2 Conclusion

In summary, this dissertation has three main contributions, all related to how to effectively and efficiently induce random forests for data series classification. These contributions have been put forward in eight publications. The conclusions of the experiments contributed towards investigating the three objectives and, by extension, contributed towards answering the overarching research question underpinning the dissertation. The main research question was posited to study how effective and efficient random forest models can be created to support the efficient and effective classification of data series,

in order to facilitate the construction of usable machine learning systems from databases characterized by large amounts of temporal data from heterogeneous sources. In short, this dissertation provided a novel investigation into the random forest algorithm and contributed key insights into its application for the purpose of inducing effective and efficient models from data series. The thesis also in part contributes towards an improved understanding of random forests for high-dimensional and sparse data.

The first objective was to investigate how random forest models can be constructed from high-dimensional and sparse representations, which are often the result of projecting discrete data series to multidimensional representations. To this end, Paper **III** contributed and evaluated a resampling-based splitting criterion for randomized decision trees which were combined into a random forest. Similarly, Paper **II** contributed an investigation of random indexing for reducing the dimensionality of high-dimensional and sparse discrete data series in conjunction with the random forest algorithm. Both of these studies contributed towards investigating the first objective, regarding the effective and efficient application of random forests for the purpose of dealing with sparse and high-dimensional data.

The second objective was to investigate how to induce efficient and effective random forests directly from homogeneous (numerical) heterogeneous (numerical and discrete) data series. Random forests for handling homogeneous numerical data series were the primary focus of most of the included studies and were investigated in depth in Paper **IV**-**VI**. A random forest for univariate numerical data series was presented and evaluated in **IV** in terms of its predictive performance. In Paper **V** an extension of the random shapelet forest was proposed to support multivariate data series by inducing one model per dimension using a split-and-combine approach. While this model was more effective than the alternatives, a more successful extension of the algorithm presented in Paper **IV** was contributed in Paper **VI** to take advantage of relations between and within the data series dimensions.

In Paper **VII**, an extension of the random forest algorithm was presented to support heterogeneous data series classification. This formulation of the algorithm was evaluated in the context of predicting adverse drug events from electronic health records. The random shapelet forest was shown in the empirical investigations to perform competitively in terms of efficiency and effectiveness compared to alternative state-of-the-art algorithms. This conclusion holds for both the task in which they were applied, i.e., for predicting adverse drug events, and in relation to alternative models when compared using benchmark datasets.

The third objective was to consider the efficient and effective induction of random forests for the purpose of providing predictions as early as possible,

in settings where the test instances are streamed one value at a time. In Paper **VIII**, an extension to the random shapelet forest introduced in Paper **IV–VI** was introduced to enable the early prediction of data series. The contributions were evaluated and the empirical investigation showed that the early random shapelet forest can provide a competitive compromise between earliness and predictive performance, when compared to related methods.

In conclusion, the overall research question in this dissertation was to investigate strategies for inducing efficient and effective random forest models from data series. The main conclusion related to the research question is that rather minor changes to the random forest learning algorithm can improve the predictive performance for high-dimensional and sparse data and that by directly handling local phase-independent similarities during the induction, even stronger models can be created for the purpose of data series classification. It was shown that one benefit of these models is that they can also be used to provide predictions early. Finally, the conclusions of this thesis not only reaffirm the empirical effectiveness of random forests for traditional multidimensional data but also indicate that the random forest framework can, with success, be extended to sequential representations. Specifically, the thesis demonstrates the importance of considering order when constructing random forest models.

5.3 Future directions

Based on the discussion and conclusions of this dissertation, there are many future directions that can be taken. Perhaps the most important direction concerns extended evaluations of the random forest proposed in Paper **VII** to handle other types of representations not limited to data series. This could, for instance, be graphs, documents or other data sources. Concerning this future direction, another important direction includes the extraction of alternative patterns from data series, perhaps consisting of wavelets, interval or other temporal patterns. Other directions for future work based on the random shapelet forest could be to investigate approaches for handling multivariate data series, for which dimensions can be missing (at random) for subsets of examples. Since having missing sensors is common in many domains, this would be important for solving many real-world data series prediction tasks.

Although the dimensionality reduction technique based on random indexing investigated in Paper **II** has been shown to yield improved predictive performance and improved efficiency, future directions could include investigating alternative strategies for creating the new dimensionalities. For example, instead of constructing a single lower dimensional space and inducing a single random forest, this process could be repeated (possibly with varying dimensionality) for each tree (or each node) in the forest and thereby increase

the variability of the individual models, similar to rotation forests [Rodriguez et al., 2006].

The random forest models introduced for data series classification do indeed, as shown by the empirical investigation, provide state-of-the-art predictive performance, for both univariate and multivariate sequences. However, one advantage of single decision trees is the possibility of interpreting the predictive model and in doing so gain insights into the predictions. For random forests, the importance of an attribute's contribution to the prediction can be computed in various ways. For the random forest introduced in Paper **IV**, a similar procedure is investigated in which the importance of regions and shapelet sizes is used to interpret the resulting model (e.g., see Figure 4.3). However, since this method is limited to globally important regions and not shapelets, one direction for future work concerns alternative approaches for interpreting the forests, e.g., by providing the importance of specific shapelets, or in the multivariate case, specific dimensions.

Bibliography

Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. (2012). *Learning from data*. AMLBook, Singapore.

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer, New York.

Allison, B., Guthrie, D., and Guthrie, L. (2006). Another look at the data sparsity problem. In *Proceedings of the 9th International Conference on Text, Speech and Dialogue*, pages 327–334. Springer.

Almenoff, J., Pattishall, E., Gibbs, T., DuMouchel, W., Evans, S., and Yuen, N. (2007). Novel statistical tools for monitoring the safety of marketed drugs. *Clinical Pharmacology & Therapeutics*, 82(2):157–166.

Alpaydin, E. (2014). *Introduction to machine learning*. MIT Press, Cambridge, MA, USA.

Amaratunga, D., Cabrera, J., and Lee, Y.-S. (2008). Enriched random forests. *Bioinformatics*, 24(18):2010–2014.

Amit, Y., Geman, D., and Wilder, K. (1997). Joint induction of shape features and tree classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 19(11):1300–1305.

Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79.

Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2016). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, pages 1–55.

Bagnall, A., Lines, J., Hills, J., and Bostrom, A. (2015). Time-series classification with COTE: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535.

Bankó, Z. and Abonyi, J. (2012). Correlation based dynamic time warping of multivariate time series. *Expert Systems with Applications*, 39(17):12814–12823.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139.

Baydogan, M. G. and Runger, G. (2014). Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, 29(2):400–422.

Baydogan, M. G. and Runger, G. (2015). Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery*, 30(2):1–34.

Baydogan, M. G., Runger, G., and Tuv, E. (2013). A bag-of-features framework to classify time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2796–2802.

Beijer, H. and De Blaey, C. (2002). Hospitalisations caused by adverse drug reactions (ADR): a meta-analysis of observational studies. *Pharmacy World and Science*, 24(2):46–54.

Biau, G. (2012). Analysis of a random forests model. *Journal of Machine Learning Research*, 13(Apr):1063–1095.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250.

Bostrom, A. and Bagnall, A. (2015). Binary shapelet transform for multiclass time series classification. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 257–269. Springer.

Boström, H. (2012). Forests of probability estimation trees. *International journal of pattern recognition and artificial intelligence*, 26(02):125–147.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC Press.

Cao, L., Chua, K. S., Chong, W., Lee, H., and Gu, Q. (2003). A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1):321–336.

Caruana, R., Karampatziakis, N., and Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 96–103. ACM.

Cetin, M. S., Mueen, A., and Calhoun, V. D. (2015). Shapelet ensemble for multi-dimensional time series. In *Proceedings of SIAM International Conference on Data Mining*, pages 307–315. SIAM.

Chen, L. and Ng, R. (2004). On the marriage of l_p-norms and edit distance. In *Proceedings of the International Conference on Very Large Data Bases*, pages 792–803. ACM.

Chen, L. and Özsu, M. T. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 491–502. ACM.

Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, 36(3):287–314.

Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. CRC Press.

Cutler, A. and Zhao, G. (2001). Pert-perfect random tree ensembles. *Computing Science and Statistics*, 33:490–497.

Dachraoui, A., Bondu, A., and Cornuéjols, A. (2014). Evaluation protocol of early classifiers over multiple data sets. In *Proceedings of the International Conference on Neural Information Processing*, pages 548–555. Springer.

Dachraoui, A., Bondu, A., and Cornuéjols, A. (2015). Early classification of time series as a non myopic sequential decision making problem. In *Proceedings of the European Conference of Machine Learning and Knowledge Discovery in Databases*, pages 433–447. Springer.

Dalianis, H., Hassel, M., Henriksson, A., and Skeppstedt, M. (2012). Stockholm EPR corpus: a clinical database used to improve health care. In *Proceedings of the Swedish Language Technology Conference*, pages 17–18.

De Condorcet, N. (1785). Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7(Jan):1–30.

Deng, H., Runger, G., Tuv, E., and Vladimir, M. (2013). A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153.

Denil, M., Matheson, D., and de Freitas, N. (2014). Narrowing the gap: Random forests in theory and in practice. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Díaz-Uriarte, R. and De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Dietterich, T. G. (2000a). Ensemble methods in machine learning. In *Proceedings of the International workshop on Multiple Classifier Systems*, pages 1–15. Springer.

Dietterich, T. G. (2000b). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. *Very Large Databases (VLDB) Journal*, 1(2):1542–1552.

Duin, R. P. (1996). A note on comparing classifiers. *Pattern Recognition Letters*, 17(5):529–536.

Esling, P. and Agon, C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res*, 15(1):3133–3181.

Friedman, J. H. (1976). A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, 26:404–408.

Friedman, J. H. (1997). On bias, variance, 0/1–loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77.

Fulcher, B. D. and Jones, N. S. (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037.

Furberg, C. D. and Pitt, B. (2001). Withdrawal of cerivastatin from the world market. *Current Controlled Trials in Cardiovascular Medicine*, 2(5):205–207.

García, S. and Herrera, F. (2008). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 9(Dec):2677–2694.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.

Geurts, P. (2001). Pattern extraction for time series classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 115–127. Springer.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220.

Goldstein, B. A., Hubbard, A. E., Cutler, A., and Barcellos, L. F. (2010). An application of random forests to a genome-wide association dataset: methodological considerations & new findings. *BMC Genetics*, 11(49).

Goldstein, B. A., Polley, E. C., and Briggs, F. B. (2011). Random forests for genetic association studies. *Statistical Applications in Genetics and Molecular Biology*, 10(1):1–34.

Grabocka, J., Schilling, N., Wistuba, M., and Schmidt-Thieme, L. (2014). Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 392–401. ACM.

Guyon, I. (1997). A scaling law for the validation-set training-set size ratio. In *AT & T Bell Laboratories*.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12:993–1001.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning [Elektronisk resurs] : Data Mining, Inference, and Prediction*. Springer New York, New York, NY, second. edition.

Hazell, L. and Shakir, S. A. (2006). Under-reporting of adverse drug reactions. *Drug Safety*, 29(5):385–396.

Hills, J., Lines, J., Baranauskas, E., Mapp, J., and Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Howard, R., Avery, A., Slavenburg, S., Royal, S., Pipe, G., Lucassen, P., and Pirmohamed, M. (2007). Which drugs cause preventable admissions to hospital? A systematic review. *British journal of clinical pharmacology*, 63(2):136–147.

Hyafil, L. and Rivest, R. L. (1976). Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17.

Jensen, P. B., Jensen, L. J., and Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395–405.

Jeong, Y.-S., Jeong, M. K., and Omitaomu, O. A. (2011). Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240.

Johansson, U., Löfström, T., and Boström, H. (2013). Random brains. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8. IEEE.

Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of the Cognitive Science Society*, volume 1.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001). Locally adaptive dimensionality reduction for indexing large time series databases. *ACM SIGMOD Record*, 30(2):151–162.

Keogh, E., Zhu, Q., Hu, B., Y., H., Xi, X., Wei, L., and Ratanamahatana, C. A. (2015). The ucr time series classification/clustering homepage.

Kitchin, R. (2014). *The data revolution: Big data, open data, data infrastructures and their consequences*. Sage.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1137–1145.

Kudenko, D. (1998). *Feature Generation for Sequence Categorization*. PhD thesis, Rutgers University, New Brunswick, NJ, USA.

Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Kuncheva, L. I. and Rodríguez, J. J. (2007). An experimental study on rotation forest ensembles. In *Proceedings of the International Workshop on Multiple Classifier Systems*, pages 459–468. Springer.

Kuncheva, L. I., Roli, F., Marcialis, G. L., and Shipp, C. A. (2001). Complexity of data subsets generated by the random subspace method: an experimental investigation. In *Proceedings of the International Workshop on Multiple Classifier Systems*, pages 349–358. Springer.

Lesh, N., Zaki, M. J., and Ogihara, M. (1999). Mining features for sequence classification. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 342–346. ACM.

Lin, J., Keogh, E., Wei, L., and Lonardi, S. (2007). Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144.

Lines, J. and Bagnall, A. (2014). Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592.

Lines, J., Davis, L. M., Hills, J., and Bagnall, A. (2012). A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–297. ACM.

Liu, J., Zhong, L., Wickramasuriya, J., and Vasudevan, V. (2009). uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675.

Liu, X. and Chen, H. (2013). Azdrugminer: an information extraction system for mining patient-reported adverse drug events in online patient forums. In *Proceedings of the International Conference on Smart Health*, pages 134–150. Springer.

Maharaj, E. A. and Alonso, A. M. (2014). Discriminant analysis of multivariate time series: Application to diagnosis based on ecg signals. *Computational Statistics & Data Analysis*, 70:67–87.

Maier, D. (1978). The complexity of some problems on subsequences and supersequences. *Journal of the ACM*, 25(2):322–336.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York.

Marteau, P.-F. (2009). Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318.

Mayer-Schönberger, V. and Cukier, K. (2013). *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.

Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U., and Hamprecht, F. A. (2011). On oblique random forests. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 453–469. Springer.

Mitchell, T. M. (1997). *Machine learning. 1997*. McGraw Hill, Burr Ridge.

Montgomery, D. C. (2008). *Design and analysis of experiments*. John Wiley & Sons.

Mori, U., Mendiburu, A., Keogh, E., and Lozano, J. A. (2016). Reliable early classification of time series based on discriminating the classes over time. *Data Mining and Knowledge Discovery*, pages 1–31.

Mueen, A., Keogh, E., and Young, N. (2011). Logical-shapelets: an expressive primitive for time series classification. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1154–1162. ACM.

Nanopoulos, A., Alcock, R., and Manolopoulos, Y. (2001). Feature-based classification of time-series data. *International Journal of Computer Research*, 10:49–61.

Nebeker, J. R., Barach, P., and Samore, M. H. (2004). Clarifying adverse drug events: a clinician's guide to terminology, documentation, and reporting. *Annals of internal medicine*, 140(10):795–801.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198.

Pariente, A., Gregoire, F., Fourrier-Reglat, A., Haramburu, F., and Moore, N. (2007). Impact of safety alerts on measures of disproportionality in spontaneous reporting databases the notoriety bias. *Drug safety*, 30(10):891–898.

Patri, O. P., Sharma, A. B., Chen, H., Jiang, G., Panangadan, A. V., and Prasanna, V. K. (2014). Extracting discriminative shapelets from heterogeneous sensor data. In *Proceedings of IEEE International Conference on Big Data*, pages 1095–1104. IEEE.

Pearl, J. (1978). On the connection between the complexity and credibility of inferred models. *International Journal of General System*, 4(4):255–264.

Pollack, J. B. (1990). Backpropagation is sensitive to initial conditions. *Complex systems*, 4:269–280.

Provost, F. and Domingos, P. (2000). Well-trained PETs: Improving probability estimation trees.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Elsevier.

Rakthanmanon, T. and Keogh, E. (2013). Fast shapelets: A scalable algorithm for discovering time series shapelets. In *Proceedings of SIAM International Conference on Data Mining*. SIAM.

Ratanamahatana, C. A. and Keogh, E. (2004). Everything you know about dynamic time warping is wrong. In *Proceedings of the 3rd Workshop on Mining Temporal and Sequential Data*, pages 22–25.

Raviv, Y. and Intrator, N. (1996). Bootstrapping with noise: An effective regularization technique. *Connection Science*, 8(3-4):355–372.

Rodríguez, J. J. and Alonso, C. J. (2004). Interval and dynamic time warping-based decision trees. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 548–552. ACM.

Rodríguez, J. J., Alonso, C. J., and Maestro, J. A. (2005). Support vector machines of interval-based features for time series classification. *Knowledge-Based Systems*, 18(4):171–178.

Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

Sahlgren, M. (2005). An introduction to random indexing. In *Proceedings of Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. In *Transactions on ASSP*, pages 43–49. IEEE.

Salzberg, S. L. (1997). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data mining and knowledge discovery*, 1(3):317–328.

Schneeweiss, S., Hasford, J., Göttler, M., Hoffmann, A., Riethling, A.-K., and Avorn, J. (2002). Admissions caused by adverse drug events to internal medicine and emergency departments in hospitals: a longitudinal population-based study. *European journal of clinical pharmacology*, 58(4):285–291.

Scornet, E., Biau, G., Vert, J.-P., et al. (2015). Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741.

Shevade, S. K. and Keerthi, S. S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253.

Shokoohi-Yekta, M., Wang, J., and Keogh, E. (2015). On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *Proceedings of SIAM International Conference on Data Mining*, pages 289–297. SIAM.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stefan, A., Athitsos, V., and Das, G. (2013). The move-split-merge metric for time series. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1425–1438.

Suzuki, A., Andrade, R. J., Bjornsson, E., Lucena, M. I., Lee, W. M., Yuen, N. A., Hunt, C. M., and Freston, J. W. (2010). Drugs associated with hepatotoxicity and their reporting frequency of liver adverse events in VigiBase™. *Drug safety*, 33(6):503–522.

Valentini, G. and Dietterich, T. G. (2004). Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *The Journal of Machine Learning Research*, 5(Dec):725–775.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., and Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309.

Wester, K., Jönsson, A. K., Spigset, O., Druid, H., and Hägg, S. (2008). Incidence of fatal adverse drug reactions: a population based study. *British journal of clinical pharmacology*, 65(4):573–579.

Wistuba, M., Grabocka, J., and Schmidt-Thieme, L. (2015). Ultra-fast shapelets for time series classification. *CoRR*, abs/1503.05018.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390.

Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C. A. (2006). Fast time series classification using numerosity reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1033–1040. ACM.

Xing, Z., Pei, J., Dong, G., and Yu, P. S. (2008). Mining Sequence Classifiers for Early Prediction. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 644–655. SIAM.

Xing, Z., Pei, J., and Keogh, E. (2010). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1):40–48.

Xing, Z., Pei, J., and Philip, S. Y. (2009). Early prediction on time series: A nearest neighbor approach. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1297–1302. Citeseer.

Xing, Z., Pei, J., and Philip, S. Y. (2012). Early classification on time series. *Knowledge and information systems*, 31(1):105–127.

Xing, Z., Pei, J., Philip, S. Y., and Wang, K. (2011). Extracting interpretable features for early classification on time series. In *Proceedings of the International Conference on Data Mining*, volume 11, pages 247–258. SIAM.

Ye, L. and Keogh, E. (2009). Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 947–956. ACM.

Ye, L. and Keogh, E. (2011). Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data mining and knowledge discovery*, 22(1-2):149–182.

Yu, L. and Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863.

Zhang, J., Ding, E. L., and Song, Y. (2006). Adverse effects of cyclooxygenase 2 inhibitors on renal and arrhythmia events: meta-analysis of randomized trials. *JAMA*, 296(13):1619–1632.

Zhao, J., Henriksson, A., Asker, L., and Bostrom, H. (2014). Detecting adverse drug events with multiple representations of clinical measurements. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 536–543. IEEE.

Zhao, J., Henriksson, A., Asker, L., and Boström, H. (2015). Predictive modeling of structured electronic health records for adverse drug event detection. *BMC medical informatics and decision making*, 15(Suppl 4):S1.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC Press.