FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Using Stacked Generalization for Anomaly Detection

Miguel Oliveira Sandim

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Carlos Soares (Ph.D.)

Co-Supervisor: Bernhard Pfahringer (Ph.D.)

September 19, 2017

Using Stacked Generalization for Anomaly Detection

Miguel Oliveira Sandim

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Dr. Luís Filipe Teixeira

External Examiner: Prof. Dr. Paulo Cortez

Supervisor: Prof. Dr. Carlos Soares

September 19, 2017

Abstract

Anomaly Detection is an important research topic nowadays, in which the intention is to find patterns in data that do not conform to expected behavior. This concept is applicable in a large number of different domains and contexts, such as intrusion detection, fraud detection, medical research and social network analysis.

Techniques that have been addressed within this topic are diverse, based on different assumptions about how anomalies manifest themselves within the data and can have different outputs (i.e. a numeric score or a labeled classification). Because of this heterogeneity, every technique is specialized in specific characteristics of the data and may only provide a limited insight on what anomalies exist in a given dataset.

Ensemble Learning is process that tries to incorporate the opinions of different learners in order to make a more pondered decision. This process has been successfully applied in the past to supervised and unsupervised learning problems and improvements in performance have been empirically observed. Stacked Generalization is one of these methods, in which a learning algorithm is used to combine the different learners.

Several state of the art Anomaly Detection techniques and datasets used throughout the literature were used in this work, which was divided in two different research studies. The first study focused on the performance and diversity of the Anomaly Detection techniques selected, while the second one focused on the application of Stacked Generalization to the techniques selected.

The first study gathered some evidence that most Anomaly Detection techniques used are *accurate* and *diverse*, therefore allowing the conditions for Stacked Generalization to be applied to this case. The second study concluded that the Stacked Generalization method guaranteed higher performance than the best Anomaly Detection technique on more than half of the datasets used. Replacing the Stacked Generalization method's meta-classifier with a simpler Majority Voting method improved the performance on even more datasets.

Possible future work could include gathering datasets with more observations and using a higher variety of Anomaly Detection techniques. This last point would likely require some implementation work, since most of the techniques referred in the literature are not implemented on general purpose programming languages.

Resumo

Deteção de Anomalias é uma área de investigação importante hoje em dia, na qual a intenção é encontrar padrões em dados que não estejam de acordo com o comportamento esperado. Este conceito é aplicável a um grande número de diferentes domínios e contextos, como deteção de intrusões, deteção de fraude, investigação médica e análise de redes sociais.

As técnicas que têm sido utilizadas nesta área são diversas, baseadas em diferentes assunções sobre como as anomalias se manifestam nos dados e podem ter diferentes resultados (uma pontuação numérica ou uma classificação). Devido a esta heterogeneidade, cada técnica é especializada em características específicas dos dados e pode apenas fornecer uma visão limitada sobre as anomalias que existem num conjunto de dados específico.

Ensemble Learning é um processo que tenta incorporar as opiniões de diferentes algoritmos de modo a potenciar uma decisão mais ponderada. Este processo tem sido aplicado com sucesso em problemas de aprendizagem supervisionada e não-supervisionada e melhorias na performance foram observadas empiricamente. *Stacked Generalization* é um destes métodos, no qual um algoritmo de aprendizagem é usado para combinar as opiniões de diferentes algoritmos.

Várias técnicas do estado de arte de Deteção de Anomalias e conjuntos de dados usados na literatura foram usados neste trabalho, que foi dividido em dois diferentes estudos de investigação. O primeiro estudo focou-se na performance e diversidade das técnicas de Deteção de Anomalias selecionadas, enquanto o segundo focou-se na aplicação de *Stacked Generalization* nas técnicas selecionadas.

O primeiro estudo revelou algumas evidências de que a maioria das técnicas de Deteção de Anomalias usadas é *exata* e *diversa*, garantindo as condições para que o *Stacked Generalization* seja aplicado a este caso. O segundo estudo concluiu que o método *Stacked Generalization* garantiu uma maior performance que a melhor técnica de Deteção de Anomalias em mais de metade dos conjuntos de dados usados. Substituindo o meta-classificador do método *Stacked Generalization* por um método *Majority Voting* simples melhorou a performance em ainda mais conjuntos de dados.

Possível trabalho futuro inclui reunir conjuntos de dados com mais observações e usar uma variedade maior de técnicas de Deteção de Anomalias. Este último ponto provavelmente requererá algum trabalho de implementação, dado que a maior das técnicas referidas na literatura não estão implementadas nas linguagens de programação comuns.

Acknowledgements

First of all I would like to thank my supervisors, Dr. Carlos Soares and Dr. Bernhard Pfahringer, for guiding me throughout this dissertation topic and pointing me in the right directions when developing this research work.

I would like to thank Daniel for all the support given and kind words in the right moments, my family and my friends João, Ana, Paula, Luís, Susana and Raquel for all their support. Without them, I know that I would not have been as successful as I was on my research.

Finally, I would like to thank Cláudio Sá, Tiago Cunha, Pedro Ribeiro, Fábio Pinto and Pedro Abreu from INESC TEC for their valuable insights, brainstorming sessions for this dissertation and kind availability to help me.

Miguel Oliveira Sandim

“Your time is limited, so don’t waste it living someone else’s life.
Don’t be trapped by dogma – which is living with the results of other people’s thinking.
Don’t let the noise of others’ opinions drown out your own inner voice.
And most important, have the courage to follow your heart and intuition.”

Steve Jobs

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

SVM	Support Vector Machine
LOF	Local Outlier Factor
COF	Connectivity-based Outlier Factor
ODIN	Outlier Detection using In-degree Number
LOCI	Location Correlation Integral
DBSCAN	Density-based Spatial Clustering of Applications with Noise
SOM	Self Organizing Maps
PCA	Principal Component Analysis
CAR	Classification Association Rules
EM	Expectaction-Maximization
CART	Classification And Regression Tree
NB	Naive Bayes
RF	Random Forest
MLP	Multilayer Perceptron
LR	Logistic Regression

Chapter 1

Introduction

Data Mining has become an important field in the modern world, given the large number of possible applications in many different domains such as marketing, medical research, computer vision, social network analysis, intrusion detection and fraud detection [Agg15]. This diverse range of applications is also explained by the increase in the volume of data stored (thanks to trends such as the *Internet of Things* [AIM10] and *Industry 4.0* [LFKFH14]) and their easy and wide distribution.

Anomaly Detection is a very specific but significant topic in this field, given the high number of domains in which it can be applied [KCBKK09]. In fact, the problem that motivates this field is a very common one and can be easily translated into this question: given a certain amount of data, is it possible to detect observations that deviate from the normal behavior of the data? This question can arise, e.g. in areas such as credit card fraud detection (where the deviant patterns can correspond to fraudulent transactions) or machine condition monitoring (in which the abnormal patterns can correspond to different vibration values of certain components belonging to an industrial machine, that might indicate a certain type of malfunction [LADVMS15]).

1.1 Motivation and Goals

The literature regarding Anomaly Detection techniques is very extensive and diverse, with a wide range of techniques that can have different outputs (either an *anomaly score* that indicates how much of a data instance in a dataset is an *anomaly*, or a label – *anomalous* or *normal*), as well as different assumptions (e.g. density based techniques have different underlying assumptions than clustering based techniques). This heterogeneity within Anomaly Detection techniques may cause different techniques to behave differently on the same dataset, which makes the task of choosing the right technique(s) for a specific domain very difficult and data-dependent.

The thesis intends to address this issue, by using several Anomaly Detection techniques at the same time and then combining their outputs into a single one. This is the idea behind Ensemble

Learning methods, which work by generating a group of models (which is designated by *ensemble*) and then combining their predictions into one. Ensemble Learning has proven to improve performance in machine learning applications such as classification, regression, time-series analysis and recommender systems [Agg17]. Ensemble Learning based solutions are also known to win various data mining competitions (the most well-known being the Netflix Prize challenge for recommender systems). More specifically this thesis will explore a Stacked Generalization method, which consists in using an extra model that *learns* the best way of combining the group of models.

Therefore this thesis intends to answer the following main research question:

- Can a Stacked Generalization method improve the performance of Anomaly Detection techniques, more specifically the performance of the best technique for a given dataset?

1.2 Outline

This document is structured as follows:

- Chapter 2 reports the current state-of-the-art in Anomaly Detection, by presenting a definition for this field and techniques used;

- Chapter 3 describes the concepts of Ensemble Learning and Stacked Generalization along with examples of techniques and applications in the context of Anomaly Detection;

- Chapter 4 presents the methodology followed throughout the experimental research of this dissertation;

- Chapter 5 summarizes the main results and findings of the application of the experimental methodology proposed in the previous chapter;

- Chapter 6 closes this dissertation by summarizing the results gathered in the context of the field, the main contributions of this work and possible future work topics.

Chapter 2

Anomaly Detection

This chapter introduces the concept of *Anomaly Detection* and includes an overview over the different types of techniques.

2.1 Definition

The following concept of Anomaly Detection is proposed, based on the one provided by Kandhari et al. [KCBKK09]:

Definition 2.1.1 (Anomaly Detection) *Anomaly Detection (also known as Outlier Detection and Outlier Analysis) corresponds to the problem of finding patterns in data that do not conform to expected behavior.*

The objective is to find instances d_o, within a dataset $\mathcal{D}$, that deviate so much from other instances that raises suspicions of being generated from a different mechanism [Haw80].

It is also important to distinguish this field from other similar ones [KCBKK09]:

- *Noise removal* and *noise accommodation*: where the goal is to detect and remove unwanted *anomalies* (which are designated by *noise*) that may affect the process of data analysis.
- *Novelty detection*: where the goal is to find anomalous patterns that were not observed before and mark them afterwards as being *normal* in the future (e.g. detecting emerging topics in social media).

A taxonomy was proposed by Kandhari et al. [KCBKK09] regarding the following aspects in this field: type of anomalies, learning mode and type of techniques (categorized according to their underlying idea and assumptions).

2.2 Type of Anomalies

Anomalies can be classified based on their nature into one of the following categories:

- *Point Anomaly:* when an individual data instance of a dataset can be considered anomalous, by comparing it with the rest of the dataset. This is the focus of the majority of the research in this field.

- *Contextual Anomaly* or *Conditional Anomaly*: When a individual data instance of a dataset can be considered anomalous when it is present in a certain context. This assumes that the dataset has attributes that can define a context (e.g. time – in time series or GPS coordinates – in spatial data). Figure 2.1 illustrates this type of anomaly with a time series dataset regarding the monthly temperature over a year: although t_1 and t_2 have both the same temperature value, t_2 is considered a contextual anomaly.

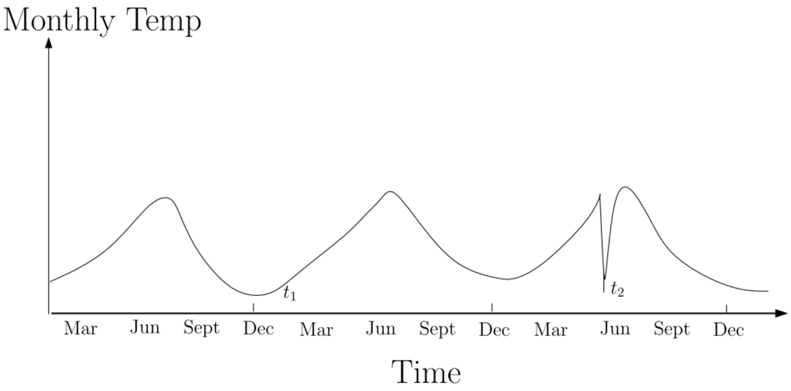

Figure 2.1: Example of one contextual anomaly (t_2) in a monthly temperature time series dataset. Source: [KCBKK09].

- *Collective Anomaly*: When a group of data instances of a dataset may not be anomalies by themselves, but when they occur together they can be considered a collective anomaly. Figure 2.2 illustrates this type of anomaly using a human electrocardiogram output time series: the red values represent a collective anomaly, although that value by itself is not considered an anomaly (despite appearing several times during the dataset just by itself).

Because of the wide scope of each of these categories, this thesis will only focus on *point anomalies* in the following sections and chapters. Information regarding the techniques capable of detecting contextual and collective anomalies can be found in Kandhari's survey on the topic ([KCBKK09]).

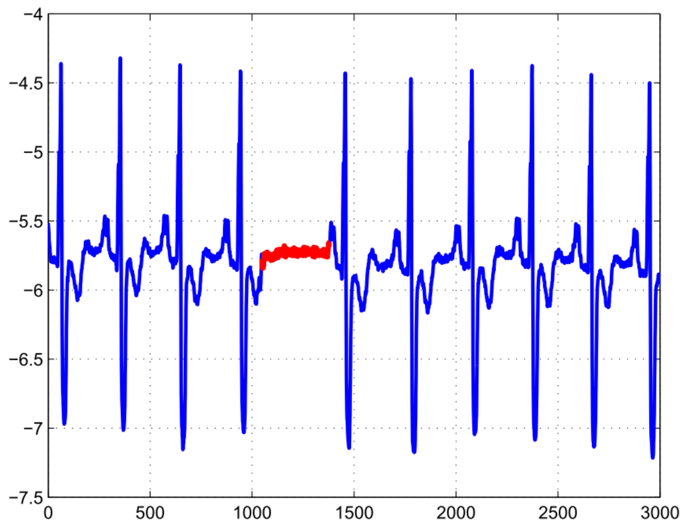

Figure 2.2: Example of one collective anomaly in a human electrocardiogram output time series dataset. Source: [KCBKK09].

2.3 Learning Mode

Anomaly Detection techniques can be classified based on the learning mode used:

- *Supervised*: Techniques using this learning mode assume that the data is fully labeled as either being *normal* or *anomalous*. Therefore, this constitutes a regular supervised learning classification problem.
- *Semi-supervised*: Techniques using this learning mode assume that the data only contains *normal* examples and try to build a model that can learn the *normal* behavior and identify examples that do not fit in this behavior. In the real-world this scenario is very frequent as in many domains it is difficult or expensive to measure anomalies and only *normal* data is available.
- *Unsupervised*: This learning mode does not require labeled data and assumes that the number of *normal* instances is much higher than the number of *anomalous* instances. Most of the Anomaly Detection techniques defined in the literature operate under this learning mode.

2.4 Type of Techniques

The techniques used in Anomaly Detection can be categorized into two groups according to their output [KCBKK09]:

- *Score output:* The techniques with this type of output assign a *score* to each data instance that represents how much the instance can be considered an anomaly. The list of anomalous instances can then be retrieved by using manually defined thresholds on the scores or by marking all the top instances as *anomalous*.

- *Label output:* The techniques that output labels resemble regular binary-classifiers in Machine Learning by either classifying a data instance as being *normal* or *anomalous*. These techniques differentiate from the *score* ones as they do not require any type of threshold definition after their application, as the data instance is already labeled as *anomalous* or *normal*.

2.4.1 Classification Based Techniques

Classification based techniques operate similarly to regular supervised learning classifiers: they train a model based on a set of labeled data and then classify each test data instance as being *normal* or *anomalous*.

One of the disadvantages of this group of techniques is that they require labeled data in the training phase of the model. Depending on the labels available in the training data, the techniques in this group can be subdivided into two types [KCBKK09]: multi-class and one-class.

2.4.1.1 Multi-class Techniques

Multi-class techniques assume that the training data contains instances belonging to several different *normal* classes and build a classifier that distinguishes each class from the remaining classes. These techniques classify a data instance as being *anomalous* if they cannot classify it as one of the *normal* classes [KCBKK09].

Examples of these techniques include certain types of Neural Networks (e.g. Multi Layered Perceptrons, Hopfield Networks), Bayesian Networks, Rule Based techniques, Decision Trees and other binary and multi-class classifiers [KCBKK09].

2.4.1.2 One-class Techniques

One-class techniques assume that the training data contains instances belonging to only one class – the *normal* one. The idea behind these techniques when learning the model is to define a decision boundary that isolates the *normal* instances. This decision boundary can therefore be used to classify new data: data instances that stay inside the decision boundry are are considered *normal* and instances that stay outside the boundary are flagged as anomalies [KCBKK09]. These techniques usually operate under the semi-supervised learning method presented in section 2.3.

Examples of these techniques include Replicator Neural Networks [HHWB], Support Vector Machines (more specifically One-class SVMs [SPSSW01]) and Rule Based techniques [KCBKK09]).

2.4.2 Nearest Neighbor Based Techniques

Nearest Neighbor based techniques are based on the assumption that *normal* data instances are situated in dense *neighborhoods* of data instances, while *anomalous* data instances situate themselves *far* from other data instances. The notion of *neighboorhoods* and *far* are employed with similarity/distance metrics that can evaluate how close (or far away) two data instances are.

These techniques can be subdivided into two different groups [KCBKK09]:

- techniques that use the distance of each data instance to its k^{th} nearest neighbor(s) as an anomaly score;
- techniques that use the concept of relative density of each data instance to compute an anomaly score (which will be detailed in this section).

2.4.2.1 Density Techniques

The assumption behind the density techniques is that a data instance that belongs to a neighborhood with low density (i.e. that contains only a few data instances) is *anomalous*, while the opposite indicates that the instance is *normal*.

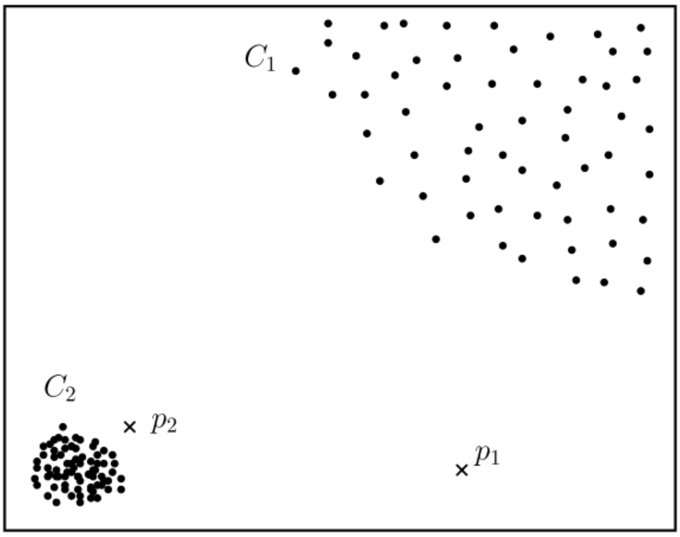

Figure 2.3: Example of a 2 dimensional dataset containing regions with different density values. Source: [KCBKK09].

However, it is important to note that this assumption may not hold if the data has regions with different density values. Figure 2.3 illustrates this example with a 2 dimensional dataset: the distance of each of the instances in cluster C_1 to their nearest neighbor is higher than the distance of p_2 to its nearest neighbor in cluster C_2. Because of this the methods that are based on this assumption would not consider p_2 as an *anomalous* instance although visually it is noticeable that this instance is *anomalous* in the given feature space.

In order to overcome this limitation, some techniques within this category compare the density of the data instances to the density of their neighbors. One of the examples of this type of techniques is the LOF (Local Outlier Factor) [BKNS00a]. Several techniques based on LOF have been proposed more recently, either to adapt this algorithm to more complex data types or to improve its efficiency. Some examples include COF (Connectivity-based Outlier Factor) [TCFC02], ODIN (Outlier Detection using In-degree Number) [HKF04] and LOCI (Local Correlation Integral) [PGF03].

2.4.3 Clustering Based Techniques

Clustering is a task in Data Mining in which the goal is to aggregate the data into meaningful or useful groups [TSK05]. Techniques that capture this idea have been applied to Anomaly Detection, out of which three groups of techniques can be distinguished in the literature based on their assumptions [KCBKK09]:

- *After clustering the data, normal data instances belong to one of the clusters formed, while anomalous data instances do not belong to any of the clusters*: several clustering algorithms (such as DBSCAN [EKSX96]) do not force all the data instances to belong to one of the clusters formed. With this particularity of the algorithms and under this assumption, we can consider these data instances as being *anomalous*.

- *Normal data instances situate themselves close to their closest cluster's centroid, while anomalous data instances remain far away from any cluster centroid*: these techniques usually use the distance of a data instance to its nearest cluster's centroid as an *anomaly score*. Examples include the use of Self-Organizing Maps (SOM) [Koh97]. It is important to note that if the *anomalous* instances form a cluster by themselves, the techniques under this assumption will not be able to detect them.

- *Normal data instances situate themselves in large and/or dense clusters, while the anomalous ones situate themselves in small and/or sparse clusters*: Examples of techniques that operate under this assumption include the FindCBLOF [HXD03].

2.4.4 Statistical Techniques

Statistical techniques operate under the assumption that *normal* data instances occur in high probability regions of a statistical model, while *anomalous* data instances occur in low probability regions [KCBKK09]. These techniques consist in building a statistical model of the data, usually using *normal* data instances, similarly to the One-class Classification techniques. However, it is important to note that these techniques have a different assumption from One-class techniques: Statistical techniques are based on statistical models and data instances are considered anomalous if they have a low probability of being generated from the learned model. One-class Classification techniques, however, are based on classification models and in the definition of a decision boundary between instances. In this case, the decision of whether a data instance is *anomalous* or not relies only in the location of the instance within the decision boundary.

The literature distinguishes parametric and non-parametric techniques, which will be detailed in this section.

2.4.4.1 Parametric Techniques

Parametric techniques are characterized by making assumptions on the distribution of the data (e.g. assuming it follows a Gaussian distribution or it can be modeled linearly) and build a statistical model of the data, by learning its parameters with *normal* data instances. The anomaly score of a new data instance can then be calculated from the probability density function of the learned model (if the instance locates itself in a region where the function has a low value it may be considered *anomalous* and vice-versa). Along with this approach, some techniques also use statistical hypothesis testing to assess if a new data instance is *anomalous* or not.

These parametric techniques can be subdivided into different groups:

- *Gaussian Model*: techniques that assume the data distribution is Gaussian. These techniques detect *anomalous* data instances based mostly on thresholds. One simple example of this type of techniques is the box plot rule [LJKLMK00].

- *Regression Model*: techniques that fit a linear model to the data. These techniques consider that a data instance is anomalous if its residual value is above a threshold. Linear models such as robust regression [LR87] have been used in these techniques.

- *Mixture of Parametric Distributions*: techniques that either model *normal* and *anomalous* data instances as belonging to two different distributions, or by modeling the *normal* data instances as belonging to a mixture of data distributions.

2.4.4.2 Non-parametric Techniques

Unlike the parametric techniques, the non-parametric approaches do not make any assumptions about the statistical distributions of the data.

These techniques, as well as the parametric ones, can be subdivided into different groups:

- *Histogram Based*: these techniques use histograms to maintain a profile of the data (usually only containing *normal* instances). The *anomaly score* of a new data instance is high if it falls in a bin of the histogram with low frequency, and vice-versa.

- *Kernel Function Based*: these techniques use kernel functions to estimate the probability distribution function, by using *normal* data instances. The *anomaly score* of a new data instance is high if it falls in a area with low probability, and vice versa.

2.4.5 Information Theoretic Techniques

Information theoretic techniques analyze the information content of the data with information theory measures (e.g. Kolomogorov Complexity, Entropy) and are based on the assumption that *anomalous* data instances induce irregularities in the information content of the data [KCBKK09].

2.4.6 Spectral Techniques

Spectral techniques are based on the assumption that *normal* and *anomalous* data instances can be distinguished in a lower feature subspace (i.e. in a new dataset with a lower number of features). These techniques often use Principal Component Analysis (PCA) [Jol02] to project the data into a lower feature space.

Chapter 3

Ensemble Learning and Stacked Generalization

This chapter provides an overview over the concepts of Ensemble Learning and Stacked Generalization.

The field of *Ensemble Learning* is presented briefly, with examples of general as well as more specific approaches used in the field of Anomaly Detection. Finally, the concept of *Stacked Generalization* is presented as well as several approaches that are used in this field.

3.1 Ensemble Learning Definition

Based on the definition provided by Mendes-Moreira et al. [MSJS12], Ensemble Learning can be defined as:

Definition 3.1.1 (Ensemble Learning) *Ensemble Learning is a process that uses a set of models (ensemble), each of them obtained by applying a learning algorithm to a given problem. This set of models is integrated in some way to obtain the final output.*

It is important to note that this definition is independent of the learning mode, which means that Ensemble Learning can be used for supervised and unsupervised learning [MSJS12]. Although Ensemble Learning is more frequently applied in supervised learning (classification and regression), it has also been used in clustering [SG03].

However, given the wide scope of these applications, this chapter and the following ones will only focus on classification applications of Ensemble Learning.

Formally, a classification model (or hypothesis) $m = (L, P, \mathcal{D})$ is an application of a learning algorithm L, with a set of defined parameters P and trained on a dataset $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, N\}$, where x_n represents the feature values of the n^{th} instance and y_n the class value of the n^{th} instance.

Given a data instance x_i from a dataset $\mathcal{D}$, $m(x_i)$ is the prediction of the class value of x_i made by model m.

Therefore, an ensemble $E = \{m_j, j = 1, \dots, J\}$ can be defined as a set of J models, where $E(x_i) = g(m_1, \dots, m_J)$ corresponds to the prediction of the class value of x_i by the ensemble E. This prediction is made using an aggregation function g which combines the predictions from the J models of the ensemble, $m_1(x_i), m_2(x_i), \dots, m_J(x_i)$. It is important to mention that this definition is recursive, as an ensemble can also be considered a model in another ensemble. The different ways in which the set of models can obtained and then integrated to obtain a final output will be discussed further in this section.

It is also important to refer that approaches with *Multiple Models* or *Multiple Learners* presented sometimes throughout the literature refer to the same concept presented in this section [MSJS12].

Dietterich [Die90] presents three reasons why Ensemble Learning can lead to better results:

- Applying a learning algorithm to a specific problem can be interpreted as searching for the best model for this problem (the one that is considered the *best* according to a predefined metric) within a space of possible models $\mathcal{H}$. When the dataset provided is too small compared to the space $\mathcal{H}$, several models can be equally considered the *best*. By building an ensemble of this set of models, it is possible to obtain a new model that may generalize better to new data.

- Some learning algorithms generate models for a specific problem by performing an optimization process over an error function, which can get stuck at a local minimum. This is the case, for example, of neural network algorithms. By building an ensemble of different models (obtained by starting this optimization at a different starting points), it is possible to obtain a model that is closer to the global minimum.

- Given a specific problem, a learning algorithm works by instantiating a model the mimics the underlying process that can explain this problem (we will represent this process by f). However, some learning algorithms (e.g. linear algorithms) may not have a model space $\mathcal{H}$ large enough to contain a model that can represent f accurately. By building an ensemble of different models and combining their outputs, it may be possible to expand the model space $\mathcal{H}$ and have a better approximation of f.

Hansen and Salamon [HS90] however state that there are two necessary (and sufficient) conditions for an ensemble of models to be more accurate than any of individual models that belong to it:

- Each of the models that compose the ensemble must be *accurate*, which according to the author is to be better than random guessing.

- The ensemble of models should be *diverse* (i.e. the outputs of the models should be uncorrelated to each other).

3.2 Ensemble Learning Process

Mendes-Moreira et al. [MSJS12] proposes three phases to be considered when using Ensemble Learning (illustrated in figure 3.1), which will be detailed in this section.

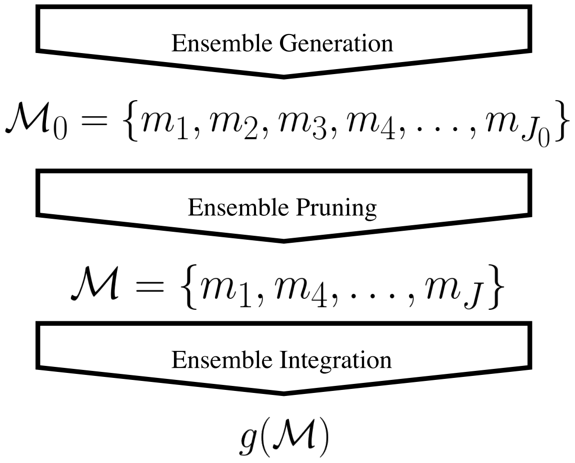
Ensemble Generation
$\mathcal{M}_0 = \{m_1, m_2, m_3, m_4, \ldots, m_{J_0}\}$
Ensemble Pruning
$\mathcal{M} = \{m_1, m_4, \ldots, m_J\}$
Ensemble Integration
$g(\mathcal{M})$

Figure 3.1: Scheme representing the Ensemble Process. Adapted from [MSJS12].

3.2.1 Ensemble Generation

The initial step in the process of Ensemble Learning is to generate an ensemble of models. We are interested in generating a set of models $\mathcal{M}_0 = \{m_j, j = 1, \ldots, J_0\}$.

Ensembles can be of two different types [MSJS12]:

- *Homogeneous*: when the set of models are generated by the same learning algorithm (e.g. tuned with different parameter settings). Most of the research work in Ensemble Learning is conducted with this type of ensembles [MSJS12].

- *Heterogeneous*: when the set of models are generated by different learning algorithms. This type of ensembles may have more diversity between models than the homogeneous type, if the nature of the learning algorithms is diverse enough [MSJS12].

It is interesting to note that homogeneous ensembles can be used in heterogeneous ensembles, given the recursive definition of an ensemble.

A possible methodology that can be followed is the *overproduce-and-choose* approach. In this methodology a high number of models are generated in the ensemble generation phase (“overproduce”), leaving the task of selecting the best models to the pruning phase (“choose”).

Mendes-Moreira et al. [MSJS12] presents different ways to produce different models in both homogeneous and heterogeneous ensembles, which will be detailed in this section.

3.2.1.1 Data Manipulation Approaches

In the definition of a model $m = (L, P, \mathcal{D})$, these approaches perform changes in the dataset $\mathcal{D}$ used to train the learning algorithm L. The same learning algorithm is trained with different datasets will result in different models (which may or may not be diverse among themselves, depending on the sensitivity of the algorithm and its sensibility to the training dataset).

Subsampling from the Training Set

This type of approach generates different models using different subsamples of the same dataset. One of the most popular approaches is *bagging* (bootstrap aggregating), which generates k subsamples of a dataset $\mathcal{D}$. These subsamples are made with replacement (a subsample can contain a data instance more than once). A model is then trained with each of the k subsamples generated, generating k different models.

Manipulating the Input Features

This type of approach can be divided in two subtypes:

- *Feature Selection*: A feature selection process is performed on the dataset, in order to generate different datasets (each one with a different subset of features). One example of this approach is the *random subspace* method [Ho98] (which chooses randomly feature subsets randomly).

- *Feature Transformation*: A transformation is conducted on the features' original values, in order to generate different datasets with different features. One example is the *input smearing* approach [FP06] that adds gaussian *noise* to each numeric feature.

Rotation forests (proposed by Rodriguez et al. [RKA06]) incorporates both feature selection and transformation processes. First, this method selects different k disjoint subsamples of features. Then, for every subsample, PCA is performed to project the feature space into a new one, where the new features correspond to linear combinations of the original ones.

3.2.1.2 Model Generation Manipulation

This type of approaches manipulates the learning algorithm's parameters or learning conditions.

Manipulating the Parameter Set

Manipulating the parameter set of a learning algorithm is a possibility to generate different models, either by iterating by ranges of possible values (Grid Search [H+03]) or using a Random Search [BB12].

Manipulating the Induction Process

In order to to obtain a model m from a learning algorithm L on a dataset $\mathcal{D}$ it is necessary to perform *induction*. This type of approaches try to change the way in which the model is generated, allowing the generation of models under different induction conditions. One of the most common approaches is to change the error function in optimization-based learning algorithms (such as neural networks).

Manipulating the Generated Model

This type of approaches performs adjustments on an already generated model, leading to different models. One known approach is to change a Classification Association Rules (CARs) model by subsampling the model's set of rules n times, generating n models with different sets of rules.

3.2.2 Ensemble Pruning

The generation of an ensemble in the previous phase, although might guarantee a wide diversity of models, it does not guarantee that the smallest ensemble possible with maximum accuracy was obtained. Several of the models may also have very correlated outputs, which do not add any extra knowledge to the final prediction. Also, since some of the approaches for generating ensembles involve randomness, there is no guarantee all the models in the ensemble will contribute positively to the final prediction.

Therefore the goal of Ensemble Pruning is to improve the predictive accuracy of the ensemble and reduce the *cost* of the ensemble (since an ensemble with a higher number of models will be more computationally costly to use).

Ensemble Pruning consists in selecting a subset $\mathcal{M}$ with J models of the set of models generated in the previous step. This phase corresponds to the "choose" step of the *overproduce-and-choose* methodology presented in section 3.2.1. Therefore:

$$\mathcal{M} \subseteq \mathcal{M}_0 \tag{3.1}$$

Mendes-Moreira et al. [MSJS12] proposes two types of approaches for conducting Ensemble Pruning, which will be detailed in this section.

3.2.2.1 Partition-Based Approaches

The main idea of partition-based approaches is to cluster the models into several groups. This could be done, for example, with the clustering algorithm k-means, in order to obtain a set of clusters of similar models. Afterwards, one or more representative models from each group are chosen to constitute the pruned ensemble.

3.2.2.2 Search-Based Approaches

Search-based approaches can divided in three different types:

- *Exponential Search Approaches*: Exponential Search Approaches search the complete search space of possible models to be included from $\mathcal{M}_0$. This search space has $2^{J_0} - 1$ possible subsets of models and the search for the optimal subset is an NP-complete problem.

- *Randomized Search Approaches*: Randomized Search Approaches perform a heuristic search in the search space (e.g. using evolutionary algorithms). Approaches such as genetic algorithms, tabu search and population-based incremental learning have been used in previous works [RG01].

- *Sequential Search Approaches*: Sequential Search Approaches perform a search by iteratively adding and/or removing a model from subset to maximize some criteria. This can be done using using *Forward Subset Selection*, *Backward Subset Selection* or a combination of both. In *Forward Subset Selection*, the search starts with am empty ensemble and models are iteratively added. In the case of *Backward Subset Selection*, the search starts with all the possible models generated in the ensemble and they are iteratively removed.

3.2.3 Ensemble Integration

The final step in Ensemble Learning is the combination of the predictions from the models in the ensemble.

In classification, the most popular approaches to combine models can be divided into two categories: combination-based approaches and model-based approaches.

3.2.3.1 Combination-based Approaches

Combination-based approaches are based on combination rules of the class values outputted by the models in the ensemble. First it is important to define the decision of the j^{th} model (referred in section 3.1 as *class value*) as $d_{j,c} \in \{0,1\}, j = 1, \ldots, J$ and $c = 1, \ldots, C$, where J is the number of models in the ensemble (as defined previously in section 3.1) and C is the number of classes. If the j^{th} model outputs class c, then $d_{j,c} = 1$ and 0 otherwise.

Majority Voting

Majority Voting has three different subtypes, in which the ensemble output corresponds to the class predicted by all classifiers (*unanimous voting*), the class predicted by at least one more than half the number of classifiers (*simple majority*) or the class predicted by the majority of the classifiers, even if it is predicted by less than half of the number of classifiers (*plurality voting*) [Pol12].

The Majority Voting approach (unless specified otherwise) usual refers to *plurality voting* [Pol12] and the decision of which class value to output can be defined as follows:

$$\arg\max_{c} \sum_{j=1}^{J} d_{j,c} \tag{3.2}$$

Weighted Majority Voting

If it is known that some of the models are more likely to make correct predictions than others, weighting the decisions of the models can improve the performance of the Majority Voting approach [Pol12]. In this case, models with higher performance would have a bigger weight assigned and models with a worse performance otherwise. We define the weight of a model m_j as w_j. These weights usually are normalized so that:

$$w_j \in [0,1] \ \wedge \ \sum_{c=1}^{C} w_j = 1, \ j = 1, \dots, J \tag{3.3}$$

In this case, the decision of the class output is defined as follows:

$$\arg\max_{c} \sum_{j=1}^{J} w_j \cdot d_{j,c} \tag{3.4}$$

A estimation of the weights could be performed by estimating the models' generalization performance in a separate validation set.

Borda Count

The Board Count method assumes that each model is capable of ranking its support to each class c and takes this into consideration [Pol12]. This method can be particularly useful in multi-class problems where C takes a considerable value.

For each model m_j, each class c receives $C - r$ votes being r the position of c in the ranking belonging to model m_i. For example, if $C = 4$ and the class 1 is ranked 3^{rd} by model m_1 (meaning that model m_1 picked class 1 as being the third most probable), then class 1 will receive $4 - 3 = 1$ votes. This procedure is then executed for each model and possible class value, the results are added up and the class with higher number of votes is chosen.

3.2.3.2 Model-based Approaches

Throughout the literature in Ensemble Learning, several more complex methods of prediction combinations are described [Pol12]. Some of these can be considered model-based, in the sense that there is a training phase of an algorithm that "learns" how to combine the several models in the ensemble. We will describe briefly two possible approaches in this section.

Stacked Generalization

Stacked Generalization (also known as Stacking) is an Ensemble Learning method in which the predictions of the models are combined using another model (also known as a *meta-classifier*) [Pol12]. In order to do so, a new dataset is generated using the prediction outputs of the models belonging to the ensemble. This new dataset is then used to generate another model (the *meta-classifier*). This mechanism is illustrated in figure 3.2.

This approach can be seen as an extension of the Weighted Majority Voting. However, unlike this method, the impact of each model in the final decision is not translated into a single value.

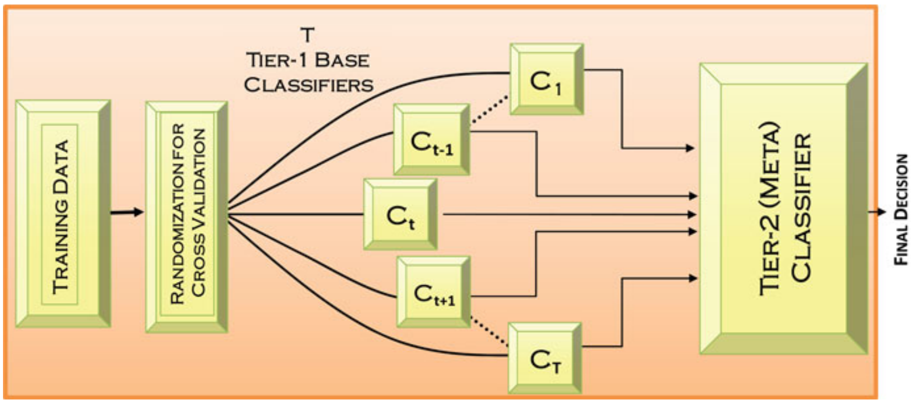

Figure 3.2: Scheme of the Stacked Generalization approach. Source: [Pol12].

Stacking determines which models are likely to be accurate in different parts of a dataset's feature space, since certain models may be more "specialized" in predicting correctly certain data instances. In this case the predictions of these models for these data instances will have a higher "weight" and the remaining models a lower one.

Since this approach is the main focus of this dissertation, we will focus on it later in this chapter.

Mixture of Experts

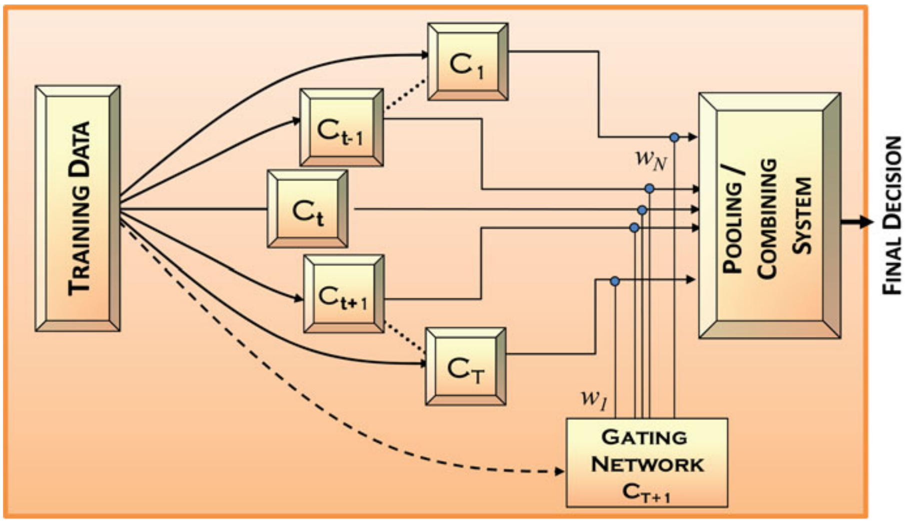

Figure 3.3: Scheme of the Mixture of Experts approach. Source: [Pol12].

As the name reflects, the Mixture of Experts approach assumes certain individual models may be *experts* in predicting the class value for certain data instances but more inaccurate for the remaining ones in the dataset. This background idea is very similar to the one behind Stacking, in which weights are assigned to each model of the ensemble reflecting its accuracy in certain parts of the dataset's feature set.

However, these weights are not determined by a new model but by a *gating network* (as illustrated in 3.3). This gating network is trained using the expectation-maximization (EM) algorithm on the original dataset.

3.3 Ensemble Learning Applications to Anomaly Detection

Ensemble Learning has been previously used with Anomaly Detection techniques [Agg17]. Because these applications were typically based on unsupervised learning, we will focus on these in this section. Several applications using Stacking (and therefore supervised learning based) will be discussed in the next section.

3.3.1 Unsupervised Learning Approaches

Aggarwal [Agg17] classifies unsupervised learning approaches as being either sequential or independent.

3.3.1.1 Sequential Approaches

In sequential approaches, several models are applied sequentially either to the entire dataset or portions of it [Agg17]. The underlying assumption of this group of approaches is that the application of each algorithm allows a more refined execution by either modifying the data or the subsequent models. Data modifications could include some of the approaches described in section 3.2.1.1, such as subsampling the dataset, performing feature selection and feature transformation [Agg17]. The final decision can be either the decision of the last applied model or a combination of the several models applied.

Models in earlier stages of a sequential approach could, for example, remove more obvious *anomalous* instances of the data so that latter models perform a more robust anomaly detection [Agg17]. The latter might then be able to have a better understanding of less-noticeable *anomalous* instances the data. This can be used, for example, with clustering based techniques, in which more robust clusters can be built after the most *anomalous* instances are removed [BLCLJ03].

3.3.1.2 Independent Approaches

In independent approaches, several models are used without having any effect on one another. These models can be applied either to the entire dataset or to portions of it [Agg17]. The underlying assumption of these approaches is that several Anomaly Detection techniques can be specialized on certain instances of the dataset, so therefore an application of these techniques and consequent combination of predictions might lead to more accurate decision. The methodologies to generate the models in these approaches include some of the ones already described in section 3.2.1, such as feature selection and dataset subsampling.

Some approaches within this category use models with the LOF ([BKNS00b]) and LOCI ([PGF03]) learning algorithms.

3.3.1.3 Ensemble Integration

One of the difficulties with unsupervised learning Anomaly Detection techniques is that they usually output a numeric score. Different techniques can output scores in different scales as, some techniques might output a normalized score (e.g. LOF), where others might output a raw distance score (e.g. k-nearest neighbor) [Agg17]. Different techniques might also have a different ordering of the scores, as some techniques output larger scores for *anomalous* instances, while others output smaller scores for this type of instances [Agg17]. Therefore, it is important to normalize the scores of each technique so that they can be meaningfully combined without over-weighting specific techniques [Agg17].

Normalization of Scores

The first step is to make sure that each model of the ensemble has the same ordering of the scores. This can be solved by flipping the sign of the scores of the models in which lower score values correspond to higher probability of being an anomalous data instance. By doing this, in every technique a higher score will always correspond to a higher probability of a data instance being anomalous.

The second step is to convert the scores of the different models into comparable values. Aggarwal [Agg17] presents two possible methods:

- *Range-based scaling*: Range-based scale uses the maximum and minimum scores of one model for a specific dataset to convert the scores. The converted scores will then lie in the interval $[0, 1]$.

  Let $s_j(x_i)$ the score that the model m_j outputs for a data instance x_i and let max_j and min_j be the maximum and minimum value respectively of the scores of model m_j for a dataset $\mathcal{D}$. The converted score of a data instance x_i with a model m_j takes the following value $s'(x_i)$:

  $$s'_j(x_i) = \frac{s_j(x_i) - min_j}{max_j - min_j} \tag{3.5}$$

  The disadvantage of this method is that the values of the converted scores will depend highly on the values of max_j and min_j. For example, in most Anomaly Detection techniques the value of max_j is attributed to the most *anomalous* data instance. In some datasets this score might be much larger than the scores of the other data instances. This phenomena can reduce drastically the discrimination of the remaining scores and reduce the ability of distinguishing which ones might be *anomalous* [Agg17].

- *Standardization*: Standardization converts the scores into standard scores (also known as Z-values).

Let μ_j and σ_j be the mean value and standard deviation respectively of the scores of model m_j for a dataset $\mathcal{D}$. The converted score of a data instance x_i with a model m_j takes the following value $s'_j(x_i)$:

$$s'_j(x_i) = \frac{s_j(x_i) - \mu_j}{\sigma_j} \tag{3.6}$$

This method however, assumes that the scores of each model m_i follow a gaussian distribution. Although this assumption rarely holds, it is reported that this method usually provides reasonably robust results [Agg17].

Another method, discussed by Gao and Tan [GT06], is to convert the techniques' scores into probabilities using the EM algorithm.

Combination of Scores

After the scores of the different techniques are normalized they can be combined. Note that the approaches discussed in section 3.2.3 can not be used, since the score is a real number and not a nominal number.

Aggarwal [Agg17] presents two possible combination methods:

- *Averaging*: The final score is computed as the mean of the scores of the different models. Therefore, a data instance x_i will have the following score:

$$\frac{\sum_{j=1}^{J} s_j(x_i)}{J} \tag{3.7}$$

- *Maximum*: The final score is computed as the maximum score across the different models. Therefore, a data instance x_i will have the following score:

$$\max_j \; s_j(x_i) \;,\; j = 1, \ldots, J \tag{3.8}$$

3.4 Stacked Generalization

Stacked Generalization (also known as Stacking) was proposed initially by Wolpert [Wol92]. It consists of an ensemble method with three steps: 1) models are generated using one or more learning processes; 2) a new dataset is generated with the predictions of those models, together with the original target variable; and 3) a new model is obtained using the new dataset containing the predictions of the previous models [SLS15]. We refer to the models in the ensemble as the *level-0 models*, their outputs as *metafeatures* and the model built with them as the *level-1 model* or *meta-classifier*.

Formally, given a dataset $\mathcal{D}^0$, Stacking first generates a set of mutually exclusive partitions of approximate size $\mathcal{D}^0_1, \ldots, \mathcal{D}^0_Z$. Then, similarly to a Z-fold cross-validation procedure, at each iteration z, the method omits the subset D_z^0 and uses the subset $D^0 - D_z^0$ as a training set to generate M level-0 models by training several learning algorithms.After the level-0 models have been generated for each iteration z, they are applied to the dataset D_z to obtain the predictions that will be used as the level-1 dataset values. We will refer to this dataset as the meta-dataset $\mathcal{D}_z^1$. The process is repeated for all Z datasets and the complete level-1 dataset, D^1 is defined as:

$$\bigcup_{z=1}^{Z} \mathcal{D}_z^0 \tag{3.9}$$

The dataset $\mathcal{D}^1$ has the same number of rows as $\mathcal{D}^0$, but M features (whose values are the predictions of the M level-0 models) plus the class value. The dataset $\mathcal{D}^1$ can then be used to train a learning algorithm, which becomes the level-1 model [SLS15].

To classify a new instance x_i, the level-0 models produce a vector of predictions $m_1(x_i), \ldots, m_M(x_i)$. This vector is the input to the level-1 model, which makes a prediction regarding the class value of x_i.

3.4.1 Applications to Anomaly Detection

The application of Stacking for Anomaly Detection is recent and sometimes not very transparent and easy to track. However, we can emphasize two approaches in the literature:

- Micenková et al. [MMA14] presented a Stacking Generalization methodology for Anomaly Detection, using outputs from two unsupervised Anomaly Detection techniques (k-NN outlier and LOF). Among with these two techniques, the authors used feature bagging which consists in a feature selection approach to generate different models from the same learning algorithms. The meta-classifier used in this approach was a model based on the Logistic Regression learning algorithm with L1 Regularization.

- Cerqueira et al. [CPSS16] proposed an approach similar to Stacking, in which the predictions from several models (LOF and Hierarchical Agglomerative Clustering) were added to the original dataset. According to our notation, the dataset used in this work has the features from $\mathcal{D}^0$ and $\mathcal{D}^1$. The meta-classifier used in this approach was a model based on the XGBoost learning algorithm.

Chapter 4

Experimental Methodology

In order to study the application of Stacking to the problem of Anomaly Detection, first several techniques were evaluated concerning their predictive performance as well as the diversity among themselves, since these are the two concepts that need to be present in order to obtain better performances using Ensemble Learning methods (as described in section 3.1). Afterwards, a study using the same techniques was conducted using Stacking approaches and their performance was analyzed.

This chapter will describe the experimental methodology followed throughout this thesis, more specifically:

- The first experimental study, focused on the performance and diversity of several Anomaly Detection techniques.

- The second experimental study on the performance of Stacking approaches using some of the Anomaly Detection techniques used on the first study.

4.1 Objectives

The original idea of this experimental research was to measure the performance and diversity of each of the techniques available and, based on these results, select the best techniques to group in an ensemble and evaluate its performance.

Given a dataset $\mathcal{D}$ divided in three mutually exclusive partitions of approximate size $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, we could separate this experimental research into three phases (as illustrated in figure 3.1):

1. *Ensemble Generation*: Select a diverse group of Anomaly Detection techniques to obtain models from.

2. *Ensemble Pruning*: Evaluate these potential models on the partition $\mathcal{D}_1$ using a cross-validation methodology (successively train a model on a set of partitions and test on a different partition) and select the top models with better performance to integrate in a ensemble $\mathcal{M}$.

3. *Ensemble Integration*: Select several possible Stacking approaches (i.e. usings a few models from $\mathcal{M}$, different meta-classifiers, ...) and evaluate each of these approaches on the partition $\mathcal{D}_2$ using a cross-validation methodology. After this evaluation is performed, the best Anomaly Detection algorithm on $\mathcal{D}_1$ and the best Stacking approach on $\mathcal{D}_2$ would then be evaluated on partition $\mathcal{D}_3$ in order to conclude if the use of Stacking could lead to a better performance.

However, given the fact that the datasets used for evaluation of this thesis do not have a large number of instances (see section 4.2.2), this division would reduce the number of instances for each of the partitions $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$.

Thus, the main premises of this work is that among the models we have available there are some that are accurate and some that are diverse. This is however a relaxation from the conditions described in section 3.1. Therefore, the methodology followed was:

1. *Experimental study on Anomaly Detection techniques*: Choose a set of Anomaly Detection techniques and examine their performance and diversity using the entire dataset $\mathcal{D}$. The goal of this study would be to evaluate if we are in the presence of at least some accurate and/or diverse models.

2. *Experimental study on Stacking approaches*: Choose a set of Stacking approaches (using different level-0 models and different level-1 models) and evaluate their performance on $\mathcal{D}$. Finally, conclude if the Stacking approaches perform better than the individual model performances.

4.2 Study on Anomaly Detection Techniques

This first study conducted during this experimental research had the following objectives:

- Study the performance and diversity of different types of Anomaly Detection techniques on several well-known datasets;

- Assess if this experimental setup contains accurate and diverse models.

4.2.1 Anomaly Detection Techniques

Techniques from several of the groups presented in chapter 2 were used in this study, more specifically Classification based techniques, Nearest Neighbor based and Clustering based. These algorithms are listed in table 4.1 and will be specified in this section according to their learning mode (i.e. supervised, semi-supervised and unsupervised). Statistical, Information Theoretic, and Spectral based techniques were not used in this study due the lack of implementations of techniques in these groups for multivariate data. A technique that predicts randomly if a data instance is *anomalous* or not was used in order to define the baseline of performance in each dataset.

Table 4.1: Anomaly Detection techniques used in this study for each nomenclature group and learning mode.

	Supervised	**Semi-Sup.**	**Unsupervised**
Classification	CART, SVM, NB, RF, MLP	One-class SVM	-
Nearest Neighbors	-	-	LOF
Clustering	-	-	DBSCAN, k-means
Statistical	-	-	-
Information Theoretic	-	-	-
Spectral	-	-	-

It is also important to point out that these techniques can have different types of outputs:

- *Binary*: A binary value indicating if a data instance is *anomalous* or not.
- *Probabilistic*: A numeric value that is always contained in the interval $[0,1]$ and can be interpreted as the probability of a data instance being *anomalous*.
- *Other Numerical*: A numerical value that is not in the interval $[0,1]$ and does not represent a probability.

The classification of the techniques according to its output is presented in table 4.2.

Table 4.2: Anomaly Detection techniques used in this study regarding the type of output.

Binary	**Probabilistic**	**Other Numerical**
One-class SVM	CART, SVM, NB, RF, MLP	k-means, DBSCAN, LOF

All the techniques's parameters were kept to the implementation's default, except for the ones in which there were no defaults. In this case, several possible values were tried for such parameters. This was the case of the techniques SVM, One-class SVM, DBSCAN, k-means and LOF. These possible values were kept as different instantions of the technique for the following reasons:

- More data would be needed in order to validate which would be the best value for each parameter of each technique;
- Some instantiations with different parameter values may be able to find *anomalous* instances other instantiations did not.

For all the algorithms an `R` implementation available was used.

4.2.1.1 Supervised

Five different supervised learning techniques were used in this study, more specifically:

- *Classification and Regression Trees (CART)*: a classification algorithm based on the tree building algorithm proposed by Breiman et al. [BFSO84];
- *Support Vector Machine (SVM)*: a classification algorithm that uses kernel functions [HDOPS98];
- *Naive Bayes (NB)*: A probabilistic classification algorithm that is based on the Bayes' theorem and assumes independence between the features [M+98];
- *Random Forest (RF)*: An ensemble learning method that trains multiple decision trees with samples of the dataset and a subgroups of the features [L+02].
- *Multilayer Perceptron (MLP)*: A feedforward artificial neural network algorithm that can have one or multiple hidden layers [R+88].

The `R` package and parameters used for each technique are detailed in table 4.3. Regarding the Random Forest technique, the default number of trees was 500 but this parameter configuration led to a very long training time in order to obtain the model. Rumelhart et al. [R+88] researched about the tuning of this parameter in 29 datasets in the context of medical data when optimizing the ROC AUC metric. The authors concluded that "from 128 trees there is no more significant difference between the forests using 256, 512, 1024, 2048 and 4096 trees". Also, "the mean and the median AUC values do not present major changes from 64 trees". Therefore, we do not believe the reduction on the number of trees in the Random Forest technique will have any significant changes in the technique's performance.

Table 4.3: Parameter values for supervised Anomaly Detection techniques.

Technique	**Parameters**	**R package**
CART	cp = 0.01	`rpart`

SVM	C = 1
	gamma = $\frac{1}{number\ of\ features}$
	kernel = {linear, polynomial (degree 3), radial, sigmoid}
		`e1071`

NB	-	`e1071`
RF	ntree = 200	`randomForest`
MLP	size = 5	`RSNNS`

The application of these techniques was automated using the `R` package `caret`.

4.2.1.2 Semi-Supervised

One semi-supervised learning technique was used in this study:

- *One-Class SVM*: Similar to the SVM technique, although this one is only trained with *normal* instances [KCBKK09].

The `R` package and parameters used for this technique are detailed in table 4.4.

Table 4.4: Parameter values for semi-supervised Anomaly Detection techniques.

Technique	**Parameters**	**R package**

One-class SVM	C = 1
	gamma = $\frac{1}{number\ of\ features}$
	kernel = {linear, polynomial (degree 3), radial, sigmoid}
		`e1071`

4.2.1.3 Unsupervised

Three different unsupervised learning algorithms were used in this study, more specifically:

- *k-means*: An approach based on the clustering algorithm k-means [R+88], in which the euclidean distance of each data instance to its closest cluster is used as an anomaly score.
- *DBSCAN*: A density-based clustering technique that has the particularity of not forcing an assignment of every data instance to a cluster [EKSX96]. Thus, instances that are assigned to a cluster may be regarded as *normal*, while the remaining ones as *anomalous*.
- *LOF*: An algorithm that detects anomalies by comparing the density of the data instances to the density of their k neighbors, where k is a parameter of the algorithm [BKNS00a]. This algorithm outputs an anomaly score for each data instance: higher scores correspond to more *anomalous* data instances.

Table 4.5: Parameter values for unsupervised Anomaly Detection techniques.

Technique	**Parameters**	**R package**
k-means	k = {3, 5, 8, 14, 19, 25, 30}	`stats`

DBSCAN	eps = {0.3, 0.5, 0.7, 0.9, 1.1}
	minPts = $number\ of\ features + 1$
		`dbscan`

LOF	k = {3, 5, 8, 14, 19, 25, 30}	`dbscan`

4.2.2 Evaluation Datasets

The datasets were gathered from an empirical study developed by Campos et al. [CZSCMSAH16], in which datasets suited for Anomaly Detection benchmarking were collected and pre-processed.

Campos et al. [CZSCMSAH16] discriminates two types of datasets used throughout the literature to benchmark Anomaly Detection techniques:

- Datasets that contain semantic information that suggests that some of the classes are sufficiently different from the remaining ones in order to be considered *anomalous* within the dataset;

- Datasets in which the *anomalous* instances are obtained by selecting a small portion of instances from a small number of classes.

The datasets are described below, including a brief description of their context as well as the mechanism that differentiates *anomalous* instances from *normal* ones. In some cases this information can not be retrieved from the literature, as each author uses different pre-processing mechanisms or different versions of a dataset and sometimes no references describing the dataset can be found.

- *ALOI*: The dataset consists in a color image collection of one-thousand small objects, recorded for scientific purposes. Several viewing angles, illumination angles and illumination colors were used for each object. Information about how *anomalous* instances were categorized was not found [CZSCMSAH16];

- *Ionosphere*: Dataset with radar data from the ionosphere. The *anomalous* instances are radar returns that show evidence of some type of structure in the ionosphere [SWHB89; CZSCMSAH16];

- *KDDCup99*: Dataset regarding Intrusion Detection events. The *anomalous* instances are the ones marked as *U2R* attacks [CZSCMSAH16];

- *PenDigits*: Dataset with pen-base handwritten digits. The *anomalous* instances are the ones classified as being the digit *4* [AA96];

- *Shuttle*: No further information regarding the context of this dataset was found. The *anomalous* instances are the ones with the class value *2* [CZSCMSAH16];

- *Waveform*: No further information regarding the context of this dataset was found. The *anomalous* instances are the ones with the class value *0* [CZSCMSAH16];

- *WBC*: Dataset composed of features extracted from digitized images of masses, in the context of breast cancer. The *anomalous* instances are the ones marked as *malignant* [CZSCMSAH16];

- *WDBC*: Dataset with similar description to *WBC*;

- *WPBC*: Dataset with similar description to *WBC*;

- *Annthyroid*: Dataset in the context of the Thyroid disease. The *anomalous* instances are the ones marked as *Hypothyroidism* [CZSCMSAH16];

- *Arrhythmia*: Dataset in the context of Arrhythmia with information regarding each patient. The *anomalous* instances are the ones marked as suffering from *Arrhythmia* [CZSCMSAH16];

- *Cardiotocography*: Dataset with features extracted from fetal cardiotocograms. The *anomalous* instances are the ones marked as the fetal state being *suspect* or *pathologic* [CZSCMSAH16];

- *HeartDisease*: Dataset in the context of heart disease with information regarding each patient. The *anomalous* instances are the ones marked having heart problems [CZSCMSAH16];

- *Hepatitis*: Dataset in the context of *Hepatitis* with information regarding each patient. The *anomalous* instances are the ones that survived [CZSCMSAH16];

- *InternetAds*: Dataset representing a set of possible advertisements on Internet pages. The features include geometry of the ad's image, phrases occurring in the URL, the image's URL, the anchor's text, and words occuring near the anchor's text [AA96]. The *anomalous* instances are the ones marked as being an ad [CZSCMSAH16];

- *PageBlocks*: Dataset representing features extracted from page layout blocks of a document [MES96]. The *anomalous* instances are the ones marked as not containing text [CZSCMSAH16];

- *Parkinson*: Dataset with features extracted from biomedical voice measurements made by patients. The *anomalous* instances are the ones marked as healthy [CZSCMSAH16];

- *Pima*: Dataset in the context of *Diabetes* with information regarding each patient. The *anomalous* instances are the ones that have *Diabetes* [CZSCMSAH16];

- *SpamBase*: Dataset with an e-mail corpus. The *anomalous* instances are the ones marked as not SPAM [CZSCMSAH16];

- *Stamps*: Dataset with color and printing properties of stamps. The *anomalous* instances are the forged stamps [CZSCMSAH16];

- *Wilt*: Dataset with image segments of land cover. The *anomalous* instances are image segments of deceased trees [CZSCMSAH16].

A characterization of the datasets used is presented in table 4.6.

Table 4.6: Number and ratio of *anomalous* and *normal* data instances in the datasets used throughout the experimental evaluation. The datasets are ordered in decreasing order by the number of outliers.

Dataset	**# Features**	**# Outliers**	**Outlier ratio**	**# Inliers**	**Inlier ratio**
ALOI	27	1508	3.04%	48026	96.96%
SpamBase	57	632	20.00%	2528	80.00%
Annthyroid	21	534	7.49%	6595	92.51%
PageBlocks	10	510	9.46%	4883	90.54%
Cardiotocography	21	412	20.00%	1648	80.00%
InternetAds	1555	368	18.72%	1598	81.28%
Wilt	5	257	5.33%	4562	94.67%
KDDCup99	40	200	0.42%	47913	99.58%
Ionosphere	32	126	35.90%	225	64.10%
Pima	8	125	20.00%	500	80.00%
Waveform	21	100	2.90%	3343	97.10%
WPBC	33	47	23.74%	151	76.26%
HeartDisease	13	37	19.79%	150	80.21%
Stamps	9	31	9.12%	309	90.88%
Arrhythmia	259	27	9.96%	244	90.04%
PenDigits	16	20	0.20%	9848	99.80%
Shuttle	9	13	1.28%	1000	98.72%
Hepatitis	19	13	16.25%	67	83.75%
Parkinson	22	12	20.00%	48	80.00%
WBC	9	10	4.48%	213	95.52%
WDBC	30	10	2.72%	357	97.28%

4.2.2.1 Data Preprocessing

All the duplicate instances (instances with the same exact values for every feature) were removed, as its existence might be problematic for some of the algorithms (e.g. LOF) [CZSCMSAH16].

Categorical features are also not universally accepted by learning algorithms. Campos et al. [CZSCMSAH16] transformed the categorical features into numeric features with the following rule: a value v (e.g. *tall*) of a categorical feature cf (e.g. *height*) was replaced by:

$$IDF(v, cf) = ln\left(\frac{N}{freq_{v,cf}}\right) \tag{4.1}$$

where N is the total number of instances in the dataset and $freq_{v,cf}$ is the number of occurrences of the value v within the categorical feature cf (e.g. number of *tall* people).

Numeric features were standardized using the formula in 3.6 (except in this case we are standardizing feature values and not scores outputted from a model).

4.2.3 Evaluation Methodology

4.2.3.1 Performance

In order to evaluate the performance of the techniques the F-measure [Pow11] was used. This metric was used instead of the ROC AUC [Pow11], since ROC AUC is usually used with numerical outputs and we have a technique with a binary output.

In order to use this metric, all the outputs were transformed into binary ones. In order to do this, for each technique the instances with a higher score value were marked as *anomalous* and the remaining ones as *normal*. The threshold for this decision was the ratio of *anomalous* instances in each dataset (e.g. if the dataset has 5% of its instances as *anomalous*, then the top 5% instances with higher score in each algorithm were predicted as *anomalous*).

The F-measure is defined as follows:

$$F_\beta = (1+\beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{4.2}$$

When $\beta = 1$, this metric is the same as the harmonic mean between *precision* and *recall*. When $\beta = 2$ or $\beta = 0.5$ this metric puts a higher weight on *recall* or *precision* respectively.

Table 4.7: Confusion matrix in the context of Anomaly Detection.

		True	
		Anomalous (Positive)	Normal (Negative)
Predicted	Anomalous (Positive)	*True positive* (TP)	*False positive* (FP)
	Normal (Negative)	*False negative* (FN)	*True negative* (TN)

Considering the definitions in table 4.7, where the positive label is *anomalous* and the negative one is *normal*, *precision* and *recall* can be defined as follows:

$$precision = \frac{TP}{TP+FP} \tag{4.3}$$

$$recall = \frac{TP}{TP+FN} \tag{4.4}$$

In the context of Anomaly Detection, *precision* can provide an insight on how many of the instances we are classifying as *anomalous* are truly *anomalous*, while *recall* on how many of all the *anomalous* instances we are classifying correctly.

The performance evaluation for the supervised and semi-supervised techniques was conducted using 10-fold stratified cross-validation. In this methodology, the dataset is divided into ten folds with equal representation of each class, where nine are used to train the model and one is used to test/evaluate the trained model. All the possible combinations of training/testing folds are used and the evaluation metric is calculated as the mean of the ones calculated for each test fold.

In the case of the unsupervised techniques there is not a training process so the technique as applied directly to the entire dataset and the evaluation was conducted on the entire dataset at once.

It is worth mentioning that sometimes the F1 metric could not be calculated: for example, when the model classifies all instances as being *normal*. In this case, the *precision* metric can not be calculated, which makes the calculation of the F1 metric impossible. In these circumstances a value of 0 was assigned to the F1 metric. This assumption penalizes this behavior heavily, which is desirable since a model that predicts all instances as *normal* is as accurate as a random guess or less.

4.2.3.2 Diversity

In order to evaluate the diversity of the outputs of the different techniques, the Jaccard metric [SST10] was used. The Jaccard metric is a similarity metric that is able to compare two binary vectors (in this case the outputs from two different techniques). The outputs of each technique were transformed into binary ones by using the method described in the previous section for the application of the F-measure.

Table 4.8: Representation of the similarity cases between two Anomaly Detection techniques A and B, where each letter a, b, c, d represents the number of occurrences for each case.

		B	
		Anomalous	Normal
A	Anomalous	a	b
	Normal	c	d

Considering the definitions in table 4.8, the Jaccard metric is defined as follows:

$$S_{Jaccard} = \frac{a}{a+b+c} \tag{4.5}$$

In the case of diversity evaluation, it would not make sense to use a cross-validation methodology. In this case the diversity metric was used on the output of each technique used to produce the level-1 dataset (as previously explained in section 3.4).

4.3 Study on Stacking Approaches

This second study conducted during this experimental research had the following objectives:

- Determine if combining several Anomaly Detection techniques with a model improves the performance of each of the Anomaly Detection techniques used in this study;

- If so, determine how much the performance is improved.

4.3.1 Stacking Approaches

The different Stacking approaches that were analyzed differ in two aspects: the Anomaly Detection techniques that were included in the ensemble and meta-classifiers used to combine the techniques in the ensemble.

4.3.1.1 Techniques Combined (Level-0)

All the techniques used in the first study were included in this study (with the values for each parameter). Additionally, the inclusion of several subgroups of techniques in the ensemble was also tried, namely:

- All of the techniques;
- Only supervised learning techniques (CART, SVM, NB, RF, MLP);
- Only semi-supervised learning techniques (One-class SVM);
- Only unsupervised learning techniques (k-means, DBSCAN, LOF);
- Only semi-supervised and unsupervised learning techniques;
- Only tree-based techniques (CART, RF);
- Only SVM-based techniques (SVM, One-class SVM);
- Only the SVM technique;
- Only clustering-based techniques (k-means, DBSCAN);
- Only the k-means technique;
- Only the DBSCAN technique;
- Only the LOF technique.

It is worth mentioning some of the techniques originated multiple models since different values were tried for some parameters. As seen previously, this is the case of the techniques SVM, One-class SVM, k-means, DBSCAN and LOF.

For each of the datasets used, the techniques with zero variance (same output for each of the instances in the dataset) were not included in the ensemble. This was done with the function `nearZeroVar` from the `caret` package, with the parameters freqCut = 0 and uniqueCut = 0.

Each of the techniques' outputs for each dataset were also standardized using the formula in equation 3.6.

4.3.1.2 Meta-classifiers (Level-1)

Several possible meta-classifiers were tried, which includes the following techniques also used at level-0: CART, MLP and RF. Additionally, the Logistic Regression (LR) technique was also used since this technique has been previously used in Stacking approaches [SLS15]. The R packages and parameters used for each technique are detailed in table 4.9.

Table 4.9: Parameter values for the meta-classifiers.

Technique	Parameters	R package
LR	maxit = 100	stats
CART	cp = 0.01	rpart
RF	ntree = 200	randomForest
MLP	size = 5	RSNNS

A Majority Voting meta-classifier was also used, to work as a baseline for the other approaches (see equation 3.2). In this case, all the outputs from the level-0 techniques were transformed into binary ones so they can be combined. This transformation is the same as the one described in section 4.2.3.1 for the application of the F-measure.

The application of these meta-classifiers was automatized using the R package caret.

4.3.2 Evaluation Methodology

The evaluation methodology was the same as the one used for the performance evaluation of the Anomaly Detection techniques (see section 4.2.3.1).

4.3.3 Evaluation Data

The evaluation data used for this study was the one described in section 4.2.2. It is worth mentioning that for the the datasets Waveform, WDBC, WPBC, Cardiotocography, HeartDisease, Hepatitis, InternetAds and Parkinson the RF technique was not used as a meta-classifier due to very long training times.

Chapter 5

Discussion of Results

This chapter will discuss the results obtained in the first and second studies, proposed in the previous chapter.

5.1 Study on Anomaly Detection Techniques

5.1.1 Performance

Table 5.1 indicates the best techniques for each dataset according to the F1 metric, along with *precision* and *recall* metrics. Different techniques (or the same technique with different parameter values) having the same F1 value for the same dataset are also listed. On 11 (52%) of the datasets the Random Forest technique was among the ones with higher F1, followed by the SVM (on 5 datasets). However, in the case of the SVM, there was not a consensus on the best value for the *kernel* parameter, which may become a disadvantage when using this technique for Anomaly Detection when compared to the Random Forest one. It is also noticeable that all the techniques in Table 5.1 are supervised learning techniques, which supports the idea that this type of techniques have superior performance compared to the semi-supervised and unsupervised ones.

Table 5.2 indicates the best semi-supervised techniques for each dataset. The values of the F1 metric for this type of techniques are considerably lower, when compared to the ones on table 5.1. However, choosing of the value of the *kernel* parameter for the One-class SVM technique appears to be easier than for the regular SVM, as on 86% of the datasets the best value was *radial*.

The results of the previous analysis for the unsupervised learning techniques are presented in table 5.3. The F1 values of this type of techniques are comparable to the ones from the semi-supervised techniques and therefore, lower that the ones on table 5.1. On 11 (52%) of the datasets, the k-means technique had the best F1 value, followed by the LOF technique (7 datasets).

In some Anomaly Detection contexts, the *precision* might be more important than *recall* and vice-versa. Therefore, an analysis on the best techniques according to the F0.5 and F2 metrics was also conducted. Table 5.4 lists the best techniques according to the F0.5 metric, in which *precision*

Table 5.1: Measurements of the metrics F1, Precision and Recall for the algorithms with highest F1 in each dataset.

Dataset	Technique	Variant	F1	Precision	Recall
ALOI	RF	-	0.590	0.600	0.580
Ionosphere	SVM	kernel = radial	0.933	0.929	0.937
KDDCup99	RF	-	0.852	0.854	0.850
PenDigits	MLP	-	1.000	1.000	1.000
Shuttle	CART	-	0.900	0.900	0.900
Waveform	SVM	kernel = radial	0.600	0.600	0.600
WBC	RF	-	0.900	0.900	0.900
WDBC	NB	-	0.900	0.900	0.900
	MLP	-	0.900	0.900	0.900
	SVM	kernel = linear	0.900	0.900	0.900
	SVM	kernel = polynomial	0.900	0.900	0.900
WPBC	SVM	kernel = linear	0.536	0.520	0.555
Annthyroid	RF	-	0.974	0.969	0.979
Arrhythmia	RF	-	0.587	0.567	0.617
Cardiotocography	RF	-	0.899	0.900	0.898
HeartDisease	NB	-	0.650	0.625	0.683
Hepatitis	RF	-	0.683	0.600	0.850
InternetAds	RF	-	0.880	0.880	0.881
PageBlocks	RF	-	0.885	0.886	0.884
Parkinson	SVM	kernel = linear	0.917	0.950	0.900
	SVM	kernel = polynomial	0.917	0.950	0.900
Pima	RF	-	0.541	0.531	0.552
SpamBase	RF	-	0.873	0.873	0.872
Stamps	MLP	-	0.886	0.800	1.000
Wilt	MLP	-	0.901	0.896	0.906

is favored. In this case, the Random Forest technique is still considered the best one on most of the datasets (11). Table 5.5 lists the best techniques according to the F2.0 metric, in which *recall* is favored. Once again, the Random Forest technique is still considered the best one on most of the datasets (11). However, in this situation na unsupervised technique (DBSCAN) is the one with better performance on two datasets (WPBC and Pima).

Regarding the comparison between the Anomaly Detection techniques and the *random* technique (that predicted every data instance randomly as *anomalous* or not, taking into account class distribution), table 5.6 lists the number of techniques (and its variants, considering the possible parameter values tested) that had a better F1 value than the *random* technique. The mean ratio of techniques with superior performance to the *random* technique among the datasets is 0.83, while the minimum is 0.53. In our experimental setup this indicates that in the worst case, only 53% of the techniques were *accurate*. However, given the fact that the mean among the datasets is considerably higher (83%), we can conclude that we have a set of techniques in which the vast majority perform better than a random guess on most of the cases. Table 5.7 reinforces this con-

Table 5.2: Measurements of the metrics F1, Precision and Recall for the semi-supervised algorithms with highest F1 in each dataset.

Dataset	**Technique**	**Variant**	**F1**	**Precision**	**Recall**
ALOI	One-class SVM	kernel = radial	0.066	0.035	0.578
Ionosphere	One-class SVM	kernel = radial	0.675	0.518	0.977
KDDCup99	One-class SVM	kernel = radial	0.016	0.008	1.000
PenDigits	One-class SVM	kernel = polynomial	0.008	0.004	1.000
Shuttle	One-class SVM	kernel = radial	0.049	0.025	1.000
Waveform	One-class SVM	kernel = radial	0.087	0.046	0.810
WBC	One-class SVM	kernel = radial	0.169	0.093	1.000
WDBC	One-class SVM	kernel = radial	0.100	0.053	1.000
WPBC	One-class SVM	kernel = polynomial	0.359	0.235	0.775
Annthyroid	One-class SVM	kernel = radial	0.203	0.116	0.809
Arrhythmia	One-class SVM	kernel = radial	0.273	0.163	0.883
Cardiotocography	One-class SVM	kernel = radial	0.464	0.312	0.910
HeartDisease	One-class SVM	kernel = radial	0.472	0.322	0.925
Hepatitis	One-class SVM	kernel = radial	0.380	0.238	1.000
InternetAds	One-class SVM	kernel = radial	0.398	0.261	0.843
PageBlocks	One-class SVM	kernel = radial	0.294	0.172	0.996
Parkinson	One-class SVM	kernel = radial	0.488	0.355	0.950
Pima	One-class SVM	kernel = radial	0.397	0.270	0.753
SpamBase	One-class SVM	kernel = radial	0.456	0.307	0.888
Stamps	One-class SVM	kernel = radial	0.282	0.165	1.000
Wilt	One-class SVM	kernel = linear	0.174	0.107	0.777

clusion, by listing the number of datasets for each technique in which the technique (or one of its variants) outperformed the *random* one. Both the Random Forest and Multilayer Perceptron techniques outperformed the *random* technique in every dataset, while the One-class SVM only did so in 71% of the datasets.

Table 5.3: Measurements of the metrics F1, Precision and Recall for the unsupervised algorithms with highest F1 in each dataset.

Dataset	Technique	Variant	F1	Precision	Recall
ALOI	LOF	k = 3	0.207	0.207	0.207
Ionosphere	k-means	k = 25	0.849	0.849	0.849
KDDCup99	k-means	k = 8	0.560	0.560	0.560
PenDigits	LOF	k = 3	0.050	0.050	0.050
	LOF	k = 5	0.050	0.050	0.050
	LOF	k = 8	0.050	0.050	0.050
	LOF	k = 14	0.050	0.050	0.050
	LOF	k = 19	0.050	0.050	0.050
Shuttle	DBSCAN	eps = 1.1	0.491	0.325	1.000
Waveform	k-means	k = 30	0.210	0.210	0.210
WBC	k-means	k = 19	0.600	0.600	0.600
WDBC	k-means	k = 30	0.700	0.700	0.700
	LOF	k = 19	0.700	0.700	0.700
WPBC	DBSCAN	eps = 0.3	0.384	0.237	1.000
	DBSCAN	eps = 0.5	0.384	0.237	1.000
	DBSCAN	eps = 0.7	0.384	0.237	1.000
	DBSCAN	eps = 0.9	0.384	0.237	1.000
	DBSCAN	eps = 1.1	0.384	0.237	1.000
Annthyroid	DBSCAN	eps = 1.1	0.190	0.122	0.438
Arrhythmia	k-means	k = 3	0.333	0.333	0.333
	LOF	k = 19	0.333	0.333	0.333
	LOF	k = 25	0.333	0.333	0.333
	LOF	k = 30	0.333	0.333	0.333
Cardiotocography	k-means	k = 3	0.364	0.364	0.364
HeartDisease	k-means	k = 14	0.351	0.351	0.351
	k-means	k = 30	0.351	0.351	0.351
Hepatitis	k-means	k = 25	0.308	0.308	0.308
	LOF	k = 25	0.308	0.308	0.308
	LOF	k = 30	0.308	0.308	0.308
InternetAds	LOF	k = 30	0.364	0.364	0.364
PageBlocks	k-means	k = 3	0.643	0.643	0.643
Parkinson	k-means	k = 8	0.667	0.667	0.667
	k-means	k = 14	0.667	0.667	0.667
Pima	DBSCAN	eps = 1.1	0.408	0.263	0.904
SpamBase	DBSCAN	eps = 1.1	0.339	0.204	0.991
Stamps	DBSCAN	eps = 1.1	0.310	0.183	1.000
Wilt	LOF	k = 5	0.152	0.152	0.152

Table 5.4: Measurements of the metrics F0.5, Precision and Recall for the semi-supervised algorithms with highest F1 in each dataset.

Dataset	Technique	Variant	F0.5
ALOI	RF	-	0.596
Ionosphere	SVM	kernel = radial	0.931
KDDCup99	RF	-	0.854
PenDigits	MLP	-	1.000
Shuttle	CART	-	0.900
Waveform	SVM	kernel = radial	0.600
WBC	RF	-	0.900
WDBC	NB	-	0.900
	MLP	-	0.900
	SVM	kernel = linear	0.900
	SVM	kernel = polynomial	0.900
WPBC	SVM	kernel = linear	0.526
Annthyroid	RF	-	0.971
Arrhythmia	RF	-	0.574
Cardiotocography	RF	-	0.900
HeartDisease	NB	-	0.634
Hepatitis	RF	-	0.628
InternetAds	RF	-	0.880
PageBlocks	RF	-	0.886
Parkinson	SVM	kernel = linear	0.933
	SVM	kernel = polynomial	0.933
Pima	RF	-	0.535
SpamBase	RF	-	0.873
Stamps	MLP	-	0.832
Wilt	MLP	-	0.898

Table 5.5: Measurements of the metrics F2, Precision and Recall for the semi-supervised algorithms with highest F1 in each dataset.

Dataset	Technique	Variant	F2
ALOI	RF	-	0.584
Ionosphere	SVM	kernel = radial	0.936
KDDCup99	RF	-	0.851
PenDigits	MLP	-	1.000
Shuttle	CART	-	0.900
	RF	-	0.900
	SVM	kernel = radial	0.900
Waveform	SVM	kernel = radial	0.600
WBC	RF	-	0.900
WDBC	NB	-	0.900
	MLP	-	0.900
	SVM	kernel = linear	0.900
	SVM	kernel = polynomial	0.900
WPBC	DBSCAN	eps = 0.3	0.609
	DBSCAN	eps = 0.5	0.609
	DBSCAN	eps = 0.7	0.609
	DBSCAN	eps = 0.9	0.609
	DBSCAN	eps = 1.1	0.609
Annthyroid	RF	-	0.977
Arrhythmia	RF	-	0.603
Cardiotocography	RF	-	0.898
HeartDisease	NB	-	0.669
Hepatitis	RF	-	0.767
InternetAds	RF	-	0.880
PageBlocks	RF	-	0.885
Parkinson	SVM	SVM_linear	0.906
	SVM	SVM_polynomial	0.906
Pima	DBSCAN	eps = 1.1	0.608
SpamBase	RF	-	0.872
Stamps	MLP	-	0.950
Wilt	MLP	-	0.904

Table 5.6: Number of techniques (and each of its variants) that had a superior value on F1 metric to the *random* technique in each dataset. The total number of techniques/variants tested was 31.

Dataset	**Variants better than random**	**Ratio**
ALOI	30	0.94
Ionosphere	29	0.91
KDDCup99	30	0.94
PenDigits	20	0.63
Shuttle	31	0.97
Waveform	31	0.97
WBC	18	0.56
WDBC	21	0.66
WPBC	17	0.53
Annthyroid	31	0.97
Arrhythmia	30	0.94
Cardiotocography	29	0.91
HeartDisease	27	0.84
Hepatitis	24	0.75
InternetAds	24	0.75
PageBlocks	30	0.94
Parkinson	29	0.91
Pima	31	0.97
SpamBase	29	0.91
Stamps	31	0.97
Wilt	18	0.56

Table 5.7: Number of datasets for which each technique had at least one variant with a better F1 value than the one from the *random* technique.

Technique	**Number of datasets**	**Ratio**
CART	19	0.90
SVM	18	0.86
NB	17	0.81
RF	21	1.00
MLP	21	1.00
One-class SVM	15	0.71
k-means	19	0.90
DBSCAN	20	0.95
LOF	20	0.95

5.1.2 Diversity

Regarding the diversity of the techniques studied, figure 5.1 displays visually the mean value of the Jaccard metric, across all the 21 datasets tested and figure 5.2 the standard deviation value. Figure 5.1 reveals visual *clusters* of similarity between several techniques:

- Supervised techniques are somewhat similar to each other but not similar to the semi-supervised and unsupervised ones;
- Semi-supervised techniques (One-class SVM) are more similar to each other and the DB-SCAN technique, but display a very low degree of similarity to other techniques;
- The LOF technique's variants are very similar to each other (this similarity is higher if the variation of the parameter *k* is lower) and somewhat similar to the k-means technique, but have a very low degree of similarity to other techniques;
- The DBSCAN technique's variants are very similar to each other (this similarity is higher if the variation of the parameter *eps* is lower) and somewhat similar to the One-class SVM variants;
- The k-means technique's variants show a medium similarity to each other and to some variants of the LOF technique.

Figure 5.2 show a more distinct variation on the similarity between the supervised learning techniques, between the DBSCAN technique's variations and between the DBSCAN's variations and one of the variations of the One-class SVM technique.

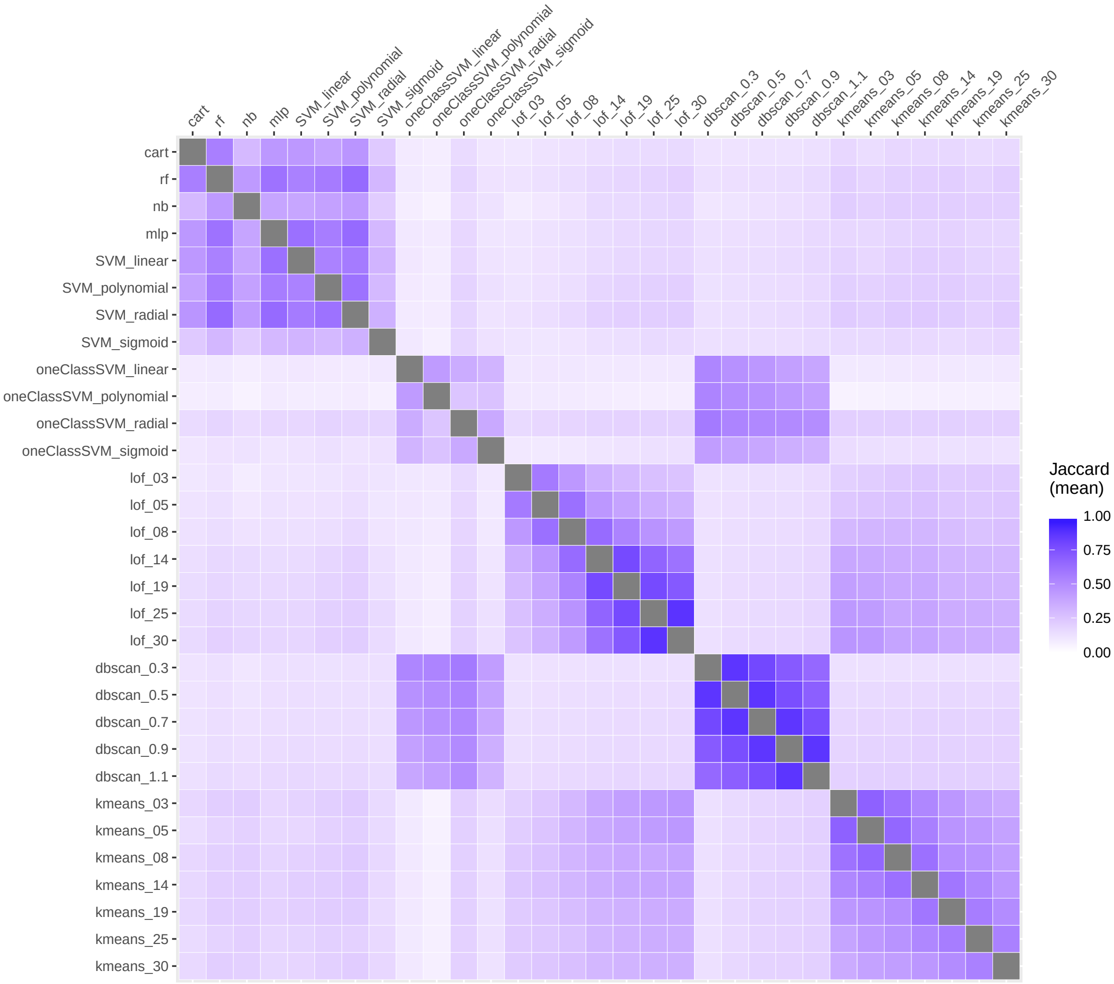

Figure 5.1: Mean value across all datasets of the Jaccard metric between each par of techniques.

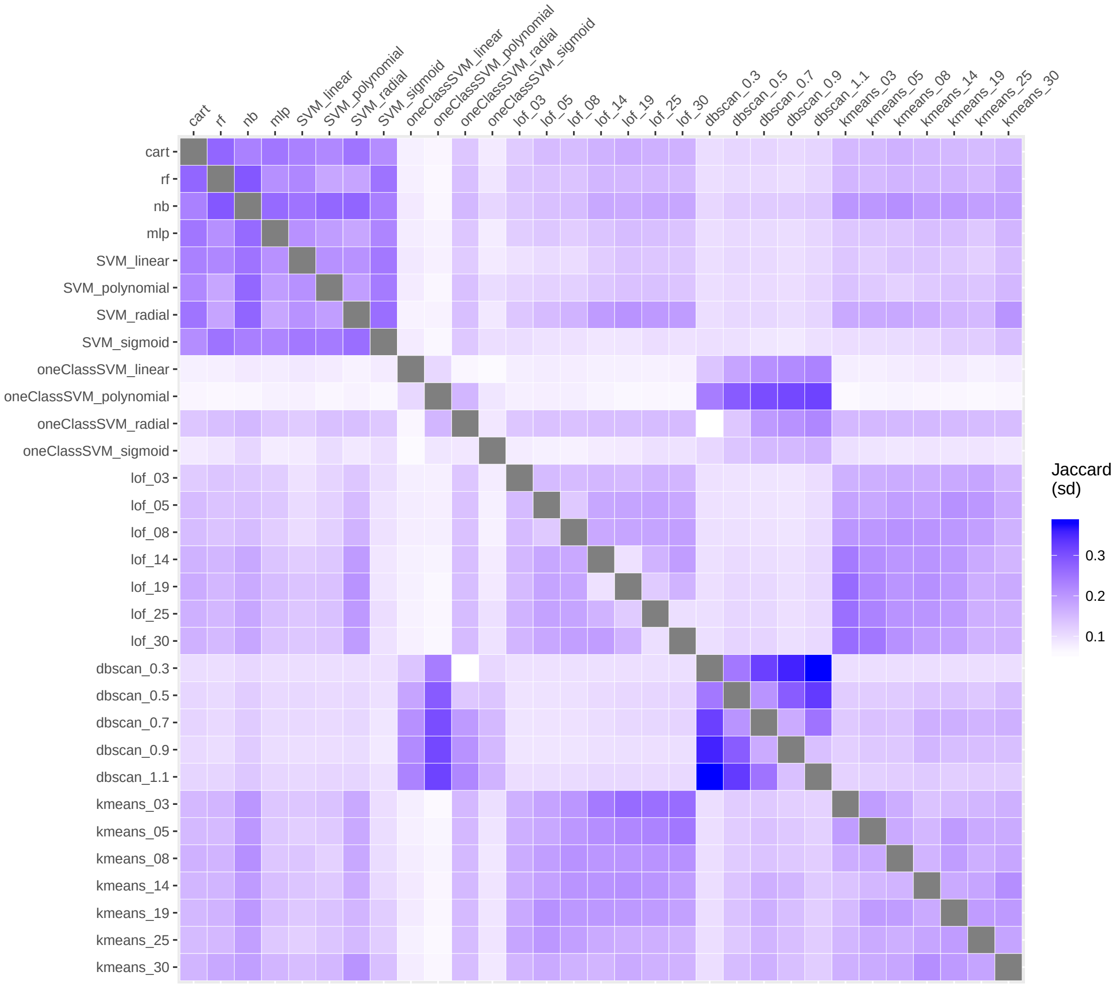

Figure 5.2: Standard deviation value across all datasets of the Jaccard metric between each par of techniques.

5.2 Study on Stacking Approaches

Table 5.8 lists the best Stacking approaches that outperformed the best single technique for each dataset. In 12 of the 21 datasets there was an improvement in the F1's value over the best technique. The mean improvement in these 12 datasets was 0.025.

Regarding the techniques used in the ensemble, the best ensemble in 6 of the 12 datasets contained techniques from several learning modes: this is the case of the datasets ALOI, KDDCup99, WPBC, InternetAds, PageBlocks and Stamps. In 5 of these 6 datasets, the best ensemble was the one that included all the techniques. Given these results, there is not a clear conclusion on whether including techniques from different learning modes results in an ensemble with higher accuracy.

Regarding the best meta-classifier, there was not one that outperformed consistently all the others on the datasets used.

Table 5.8: Measurements of the metric F1 for the Stacking approaches that outperform the best algorithm for each dataset.

Dataset	**Ensemble Techniques**	**Meta-classifier**	**Best Technique F1**	**Best Ensemble F1**	**Improvement**
ALOI	All	RF	0.933	0.947	+0.014
KDDCup99	All	RF	0.852	0.879	+0.027
Shuttle	One-class SVM + SVM	LR	0.900	1.000	+0.100
	One-class SVM + SVM	MLP	0.900	1.000	+0.100
	SVM	LR	0.900	1.000	+0.100
	SVM	MLP	0.900	1.000	+0.100
Waveform	SVM	CART	0.600	0.640	+0.040
WPBC	One-class SVM + SVM	MLP	0.536	0.569	+0.033
Annthyroid	All Supervised	CART	0.974	0.976	+0.002
Cardiotocography	All Supervised	CART	0.899	0.905	+0.006
HeartDisease	CART + RF	MLP	0.650	0.672	+0.022
InternetAds	All	LR	0.880	0.898	+0.018
PageBlocks	All	RF	0.885	0.890	+0.005
SpamBase	All Supervised	RF	0.873	0.882	+0.009
Stamps	All	CART	0.886	0.906	+0.020

Table 5.9 lists the best Stacking approaches that outperformed the best single technique for each dataset, but this time considering additionally Majority Voting as an alternative to a meta-classifier. Majority Voting is not a meta-classifier, so therefore a solution with it cannot be considered an application of the Stacking method. However, the inclusion of Majority Voting can provide insight on whether how well the meta-classifiers are performing, by serving as a baseline.

In this case, in 15 of the 21 datasets there was an improvement of the F1's value over the best technique: more 3 datasets than without considering Majority Voting. The mean value of this improvement was 0.056, also higher than the one from the previous experiment. The techniques included in most of the successful ensembles are the RF and CART algorithms, with Majority Voting being a better alternative to a meta-classifier in all datasets except Waveform and Stamps.

Table 5.9: Measurements of the metric F1 for the Stacking approaches that outperform the best algorithm for each dataset, when considering Majority Voting as an alternative to a meta-classifier.

Dataset	**Ensemble Techniques**	**Meta-classifier**	**Best Technique F1**	**Best Ensemble F1**	**Improvement**
ALOI	CART + RF	Majority Voting	0.590	0.742	+0.152
Ionosphere	DBSCAN	Majority Voting	0.933	1.000	+0.067
KDDCup99	CART + RF	Majority Voting	0.852	0.892	+0.040
Shuttle	All Supervised	Majority Voting	0.900	1.000	+0.100
	One-class SVM + SVM	LR	0.900	1.000	+0.100
	One-class SVM + SVM	MLP	0.900	1.000	+0.100
	SVM	Majority Voting	0.900	1.000	+0.100
	SVM	LR	0.900	1.000	+0.100
	SVM	MLP	0.900	1.000	+0.100
Waveform	SVM	CART	0.600	0.640	+0.040
WPBC	CART + RF	Majority Voting	0.536	0.625	+0.089
Annthyroid	CART + RF	Majority Voting	0.974	0.979	+0.005
Arrhythmia	CART + RF	Majority Voting	0.587	0.653	+0.066
Cardiotocography	CART + RF	Majority Voting	0.899	0.928	+0.029
HeartDisease	CART + RF	Majority Voting	0.650	0.676	+0.026
InternetAds	All Supervised	Majority Voting	0.880	0.913	+0.032
PageBlocks	CART + RF	Majority Voting	0.885	0.919	+0.034
Pima	CART + RF	Majority Voting	0.541	0.647	+0.106
SpamBase	CART + RF	Majority Voting	0.873	0.908	+0.036
Stamps	All	CART	0.886	0.906	+0.020

Chapter 6

Conclusions and Future Work

This chapter presents the main conclusions of this research work in the context of Anomaly Detection and Ensemble Learning and possible future work topics.

6.1 Main Overview and Conclusions

On this dissertation, we first discussed the concepts of Anomaly Detection and Ensemble Learning. A taxonomy and applications for both of the fields was also presented, among with a definition of the Stacked Generalization method and its applications in the Anomaly Detection context.

We then proposed an experimental methodology, separated in two experimental studies, to tackle the application the Stacked Generalization method in the context of Anomaly Detection. Several Anomaly Detection techniques from different taxonomic groups were studied separately and combined with different meta-classifiers. These studies were supported by datasets used throughout the literature of Anomaly Detection. The main results and findings of this experimental methodology were also exposed.

We can briefly summarize the main conclusions of this dissertation as follows:

- Most of the Anomaly Detection techniques used in this study are *accurate* and *diverse* in the datasets used, therefore having the necessary conditions for the Stacking method overperforming the best technique in each dataset.

- The application of the Stacking method guaranteed higher F1 values than the best Anomaly Detection technique on more than half of the datasets used.

- There is no clear indication whether including Anomaly Detection techniques from different learning modes guarantees higher F1 values. In the datasets where this was true, the best combination was including techniques from all the learning modes available.

- There is not a meta-classifier that clearly outperformed the others in terms of F1 on the datasets, so choosing the appropriate one seems to be very dependent on the dataset.

- Replacing the meta-classifier with the Majority Voting method improved the F1 value on even more datasets, with also a higher mean improvement on the F1. In this case, ensembles with tree-based Anomaly Detection techniques only (CART and Random Forest) were the ones with higher F1 values on most datasets.

6.2 Main Contributions

The main contribution of this dissertation is the the development of a research study on Stacking approaches applied to Anomaly Detection with a broader variety of techniques, meta-classifiers and datasets. Also a study on the performance and diversity of these techniques across datasets used throughout the Anomaly Detection literature was also provided as a secondary contribution.

6.3 Future Work

Several ideas can be followed as future work to the research work developed in this dissertation:

- *Bigger datasets*: One of the limitations of the experimental methodology proposed was the size of the datasets used. Although from our point of view it is important to perform a study on datasets that were used previously in the literature in order to enable further comparisons, these datasets are in general very small. More complex methodologies with processes like parameter optimization would need bigger datasets in order to perform validation in a greater number of data instances.

- *Higher variety of Anomaly Detection techniques*: Although this dissertation performed an empirical evaluation on Stacking with a greater variety of Anomaly Detection techniques than previous studies (at least, to the best of our knowledge), more techniques could be incorporated and tested. In particular, there were taxonomic groups of Anomaly Detection techniques that were not explored, mostly because the application of these techniques is not as popular, and therefore there is a lack of implementations in general purpose languages. Therefore, the exploration of this future work idea most probably implicate some implementation work.

References

[AA96] Fevzi Alimoglu and Ethem Alpaydin. Methods of Combining Multiple Classifiers Based on Different Representations for Pen-based Handwritten Digit Recognition. In *Proceedings of the fifth turkish artificial intelligence and artificial neural networks symposium (tainn 96*, 1996.

[Agg15] Charu C Aggarwal. *Data Mining: The Textbook*. Springer Publishing Company, Incorporated, 2015. ISBN: 978-3-319-14142-8.

[Agg17] Charu C Aggarwal. *Outlier Analysis*. Springer Publishing Company, Incorporated, 2017. ISBN: 1461463955, 9781461463955.

[AIM10] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(Feb):281–305, 2012.

[BFSO84] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[BKNS00a] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: Identifying Density-Based Local Outliers. *Proceedings of the 2000 acm sigmod international conference on management of data*:1–12, 2000. ISSN: 01635808. DOI: 10.1145/335191.335388.

[BKNS00b] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: Identifying Density-based Local Outliers. *Sigmod rec.*, 29(2):93–104, May 2000. ISSN: 0163-5808. DOI: 10.1145/335191.335388. URL: http://doi.acm.org/10.1145/335191.335388.

[BLCLJ03] Daniel Barbará, Yi Li, Julia Couto, Jia-Ling Lin, and Sushil Jajodia. Bootstrapping a Data Mining Intrusion Detection System. In *Proceedings of the 2003 acm symposium on applied computing*. In SAC '03. ACM, New York, NY, USA, 2003, pages 421–425. ISBN: 1-58113-624-2. DOI: 10.1145/952532.952616. URL: http://doi.acm.org/10.1145/952532.952616.

[CPSS16] Vítor Cerqueira, Fábio Pinto, Claudio Sá, and Carlos Soares. *Combining Boosted Trees with Metafeature Engineering for Predictive Maintenance*. In. *Advances in intelligent data analysis xv: 15th international symposium, ida 2016, stockholm, sweden, october 13-15, 2016, proceedings*. Henrik Boström, Arno Knobbe, Carlos Soares, and Panagiotis Papapetrou, editors. Springer International Publishing, Cham, 2016, pages 393–397. ISBN: 978-3-319-46349-0. DOI: 10.1007/978-3-319-46349-0_35. URL: http://dx.doi.org/10.1007/978-3-319-46349-0%7B%5C_%7D35.

[CZSCMSAH16] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, 30(4):891–927, July 2016. ISSN: 1573756X. DOI: 10.1007/s10618-015-0444-8. URL: http://link.springer.com/10.1007/s10618-015-0444-8.

[Die90] Thomas G Dietterich. Ensemble Methods in Machine Learning. *First international workshop on multiple classifier systems*, 1857:1–15, 1990.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the second international conference on knowledge discovery and data mining*. In KDD'96. AAAI Press, 1996, pages 226–231.

[FP06] Eibe Frank and Bernhard Pfahringer. *Improving on Bagging with Input Smearing*. In. *Advances in knowledge discovery and data mining: 10th pacific-asia conference, pakdd 2006, singapore, april 9-12, 2006. proceedings*. Wee-Keong Ng, Masaru Kitsuregawa, Jianzhong Li, and Kuiyu Chang, editors. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pages 97–106. ISBN: 978-3-540-33207-7. DOI: 10.1007/11731139_14. URL: http://dx.doi.org/10.1007/11731139%7B%5C_%7D14.

[GT06] Jing Gao and Pang-Ning Tan. Converting Output Scores from Outlier Detection Algorithms into Probability Estimates. In *Proceedings of the sixth international conference on data mining*. In ICDM '06. IEEE Computer Society, Washington, DC, USA, 2006, pages 212–221. ISBN: 0-7695-2701-9. DOI: 10.1109/ICDM.2006.43. URL: http://dx.doi.org/10.1109/ICDM.2006.43.

[H+03] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[Haw80] Douglas M Hawkins. *Identification of outliers*. Volume 11. Springer, 1980.

[HDOPS98] M A Hearst, S T Dumais, E Osuna, J Platt, and B Scholkopf. Support vector machines. *Ieee intelligent systems and their applications*, 13(4):18–28, July 1998. ISSN: 1094-7167. DOI: 10.1109/5254.708428.

[HHWB] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier Detection Using Replicator Neural Networks.

[HKF04] Ville Hautamäki, Ismo Kärkkäinen, and Pasi Fränti. Outlier Detection Using k-Nearest Neighbour Graph. *Proceedings of the pattern recognition, 17th international conference on (icpr'04) volume 3 - volume 03*:430–433, 2004. DOI: 10.1109/ICPR.2004.671.

[Ho98] Tin Kam Ho. The random subspace method for constructing decision forests. *Ieee transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[HS90] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *Ieee transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[HXD03] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering Cluster-based Local Outliers. *Pattern recogn. lett.*, 24(9-10):1641–1650, 2003. ISSN: 0167-8655. DOI: 10.1016/S0167-8655(03)00003-5.

[Jol02] I T Jolliffe. Principal Component Analysis, Second Edition. *Springer series in statistics*. Springer Series in Statistics 98:487, 2002. ISSN: 15364844. DOI: 10.1007/b98835.

[KCBKK09] Rupali Kandhari, Varun Chandola, Arindam Banerjee, Vipin Kumar, and Rupali Kandhari. Anomaly detection. *Acm computing surveys*, 41(3):1–6, 2009. ISSN: 03600300. DOI: 10.1145/1541880.1541882.

[Koh97] Teuvo Kohonen, editor. *Self-organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997. ISBN: 3-540-62017-6.

[L+02] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomForest. *R news*, 2(3):18–22, 2002.

[LADVMS15] Rocco Langone, Carlos Alzate, Bart De Ketelaere, Jonas Vlasselaer, Wannes Meert, and Johan A K Suykens. LS-SVM based spectral clustering and regression for predicting maintenance of industrial machines. *Engineering applications of artificial intelligence*, 37:268–278, 2015. ISSN: 09521976. DOI: 10.1016/j.engappai.2014.09.008.

[LFKFH14] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239, 2014.

[LJKLMK00] Jorma Laurikkala, Martti Juhola, Erna Kentala, N Lavrac, S Miksch, and B Kavsek. Informal identification of outliers in medical data. In *Fifth international workshop on intelligent data analysis in medicine and pharmacology*. Volume 1, 2000, pages 20–24.

[LR87] Annick M Leroy and Peter J Rousseeuw. Robust regression and outlier detection. *Wiley series in probability and mathematical statistics, new york: wiley, 1987*, 1987.

[M+98] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *Aaai-98 workshop on learning for text categorization*. Volume 752. Madison, WI, 1998, pages 41–48.

[MES96] Donato Malerba, Floriana Esposito, and Giovanni Semeraro. *A Further Comparison of Simplification Methods for Decision-Tree Induction*. In. *Learning from data: artificial intelligence and statistics v*. Doug Fisher and Hans-J. Lenz, editors. Springer New York, New York, NY, 1996, pages 365–374. ISBN: 978-1-4612-2404-4. DOI: 10.1007/978-1-4612-2404-4_35. URL: http://dx.doi.org/10.1007/978-1-4612-2404-4%7B%5C_%7D35.

[MMA14] Barbora Micenková, Brian McWilliams, and Ira Assent. Learning Outlier Ensembles : The Best of Both Worlds – Supervised and Unsupervised. *Proc. of the acm sigkdd workshop on outlier detection and description, odd.*:1–4, 2014.

[MSJS12] João Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire De Sousa. *Ensemble approaches for regression*. Volume 45(1). ACM, 2012, pages 1–40. ISBN: 3512250815. DOI: 10.1145/2379776.2379786.

[PGF03] Hiroyuki Papadimitriou, Spiros Kitagawa, Phillip B Gibbons, and Christos Faloutsos. LOCI: Fast outlier detection using the local correlation integral. *Proceedings of the icde03*, 2003.

[Pol12] Robi Polikar. Ensemble learning. In, *Ensemble machine learning*, pages 1–34. Springer US, Boston, MA, 2012. ISBN: 1441993258. DOI: 10.1007/978-1-4419-9326-7_1. URL: http://link.springer.com/10.1007/978-1-4419-9326-7%7B%5C_%7D1.

[Pow11] David Martin Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation, 2011.

[R+88] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[RG01] Dymitr Ruta and Bogdan Gabrys. *Application of the Evolutionary Algorithms for Classifier Selection in Multiple Classifier Systems with Majority Voting*. In. *Multiple classifier systems: second international workshop, mcs 2001 cambridge, uk, july 2–4, 2001 proceedings*. Josef Kittler and Fabio Roli, editors. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pages 399–408. ISBN: 978-3-540-48219-2. DOI: 10.1007/3-540-48219-9_40. URL: http://dx.doi.org/10.1007/3-540-48219-9%7B%5C_%7D40.

[RKA06] J J Rodriguez, L I Kuncheva, and C J Alonso. Rotation Forest: A New Classifier Ensemble Method. *Ieee transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, October 2006. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2006.211.

[SG03] Alexander Strehl and Joydeep Ghosh. Cluster Ensembles — a Knowledge Reuse Framework for Combining Multiple Partitions. *J. mach. learn. res.*, 3:583–617, March 2003. ISSN: 1532-4435. DOI: 10.1162/153244303321897735. URL: http://dx.doi.org/10.1162/153244303321897735.

[SLS15] M. Paz Sesmero, Agapito I. Ledezma, and Araceli Sanchis. Generating ensembles of heterogeneous classifiers using Stacked Generalization. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 5(1):21–34, January 2015. ISSN: 19424795. DOI: 10.1002/widm.1143. URL: http://doi.wiley.com/10.1002/widm.1143.

[SPSSW01] Bernhard Schölkopf, John C Platt, John C Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural comput.*, 13(7):1443–1471, July 2001. ISSN: 0899-7667. DOI: 10.1162/089976601750264965. URL: https://doi.org/10.1162/089976601750264965.

[SST10] Choi Seung-Seok, Cha Sung-Hyuk, and Charles C Tappert. A Survey of Binary Similarity and Distance Measures. *Journal of systemics, cybernetics & informatics*, 8(1):43–48, 2010. ISSN: 16904524. DOI: 10.1.1.352.6123. URL: http://www.iiisci.org/journal/CV%7B%5C$%7D/sci/pdfs/GS315JG.pdf%20http://ezproxy.uthm.edu.my/login?url=http://search.ebscohost.com/login.aspx?direct=true%7B%5C&%7Ddb=aph%7B%5C&%7DAN=59856128%7B%5C&%7Dsite=ehost-live%7B%5C&%7Dscope=site.

[SWHB89] Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. Classification of radar returns from the ionosphere using neural networks. *Johns hopkins apl technical digest*, 10(3):262–266, 1989.

[TCFC02] Jian Tang, Zhixiang Chen, Ada Wai-chee Fu, and David W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. *Advances in knowledge discovery and data mining*, 2336:535–548, 2002. ISSN: 16113349. DOI: 10.1007/3-540-47887-6.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Chap 8 : Cluster Analysis: Basic Concepts and Algorithms. *Introduction to data mining*:Chapter 8, 2005. ISSN: 00224405. DOI: 10.1016/0022-4405(81)90007-8.

[Wol92] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.