

DETECT ANOMALOUS BEHAVIOUR IN COMPUTER SYSTEMS

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

kollepara Vinay Kumar (Reg. No. 37110350)

Kasse Jaydeep (Reg. No. 37110330)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC JEPPIAAR NAGAR, RAJIV
GANDHI SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **KOLLEPARA VINAY KUMAR (Reg. No. 37110350)** and **KASSE JAYADEEP (Reg. No. 37110330)** who carried out the project entitled "**DETECT ANOMALOUS BEHAVIOUR IN COMPUTER SYSTEMS**" under my supervision from **August 2020 to March 2021**.

Internal Guide

Mrs. K.Indra.M.Tech.,

Head of the Department

Dr. S. Vigneswari, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Kollepara Vinay Kumar** and **Kasse Jaydeeeep** here by declare that the Project Report entitled “**DETECT ANOMALOUS BEHAVIOUR IN COMPUTER SYSTEMS**” is done by me under the guidance of **Mrs.K.Indra.M.Tech.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. K. Indra.M.Tech.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Anomaly detection is the process of finding outliers in a given dataset. Outliers are the data objects that stand out amongst other objects in the dataset and do not conform to the normal behavior in a dataset. Anomaly detection is a data science application that combines multiple data science tasks like classification, regression, and clustering. The target variable to be predicted is whether a transaction is an outlier or not. Since clustering tasks identify outliers as a cluster, distance-based and density-based clustering technology can be used in anomaly detection tasks. The ever-lasting challenge of detecting and mitigating failures in computer networks has become more essential than ever; especially with the enormous number of smart devices that get connected to all sorts of network everyday. Whether the root cause of a given anomaly is a security breach, a component failure, an environmental factor, or even any combination of these reasons, anomalies need to be detected and mitigated timely and properly. In this paper, we review and evaluate the state-of-the-art studies on the problem of anomaly detection in computer networks. We provide an elaborate description of the anomaly detection problem, and depict the different categorizations of its solutions. We also illustrate some recent state-of-the-art solutions on the system level, and depict current trends in handling malware-induced anomalies in computer systems. Additionally, we evaluate the presented solutions and highlight their shortcomings. The system is host-based and monitors computer network host users. The system contains an automatic anomaly detection component. This component uses a test based on the self-organizing map to test if user behavior is anomalous.

Table of content

1. Introduction	1
Problem description	2
Purpose	3
2. Anomaly detection	3
Background	4
Research area	5
Machine Learning	5
Characteristics	5
Application domain	6
3. Method	7
Software development method	7
R for statistical computing	8
Problem solving approach	9
Problem characteristics and challenges	10
4. The data analysis process	12
Data preparation	13
Data selection	13
Data cleaning	16
Data encoding	16
Text mining	16
Feature reduction	18
Mutual Information (MI)	18
Principal Component Analysis (PCA)	19
Model generation	19
The Adaboost method	20
The Support Vector Machines (SVMs) method	24
Model selection and assessment	26
5. Results and discussion	27

Prepared data	28
Reduced number of features	29
Modelling	32
Adaboost	33
SVMs	37
Selection of model	41
Final model	41
Importance of features	41
The final assessment	43
Uncertainties with the classification results	45
Anomaly score	45
Evaluation and application	47
6. Conclusion	47
Further research	48
References	50

Abbreviation

DAC	Development Assistance Committee
IATI	International Aid Transparency Initiative
MI	Mutual Information
NA	Not Available
OECD	Organization for Economic Co-operation and Development
PBA	Programme Based Approach
PCA	Principal Component Analysis
PC	Principal Component
PO	Policy Objective
Sida	Swedish International Development Cooperation Agency
SVMs	Support Vector Machines
UN	United Nations
WFM	Word Frequency Matrix

1. Introduction

The world is undergoing a digital data revolution. Every day, people around the world create 2.5 quintillion bytes of data - so much that 90% of the data in the world today has been created in the last two years alone (IBM, 2015). This prevalence of data in all that we do is changing society, organizations and individual behaviour worldwide. To thrive in this new environment, new strategies and technologies are required. Especially since most data today are underused, even though they contain latent information that could lead to important knowledge. Almost every business, governmental, and organizational activity is driven by data (Atkinson, 2013). A report from the McKinsey Global Institute asserts that Machine Learning, data mining and predictive analytics will be the drivers of the next big wave of innovation (Manyika et al., 2011). The Aid community is one of the areas that currently is adapting to the new rich data climate. International Aid Transparency Initiative (IATI) is an example of how this works in practice. IATI is a voluntary, multi-stakeholder initiative that seeks to make information about aid spending easier to access, use, and understand. This is done in order to increase the effectiveness of aid and humanitarian resources in tackling poverty. The core of the initiative is the IATI Standard, which is a format and framework for publishing data on development cooperation activities. The standard is intended to be used by all organizations in development including government donors, private sector organizations, and national and international non-governmental organizations. The Swedish International Development Cooperation Agency (Sida) is one of the stakeholders that cooperate with IATI (IATI,2015).

Sweden is a world-leading provider of development aid. One percentage of Swedish gross national income is used for development aid, which is beyond the UN-goal of 0.7% (Sida, 2014). Sida is responsible for the main share of these payments and provides support to several thousand projects in developing countries. In order to keep track of the different types of support, all contributions are classified against several statistical codes. A contribution is the financial support given by Sida for a project.

Sida's contributions can finance both part of projects and an entire project. In the list below you will find three examples of what a contribution might be:

- Project support to female leadership in Middle East implemented through an international non-governmental organization.

- Core support to a partner country public organization.

- Support to a program aimed at developing trade and environment in Guatemala.

A contribution can consist of one or more components, but usually a

contribution consists of just one component. Statistical information is linked to

the component. The statistics describe the characteristics of a component, for example which sector it belongs to and what kind of organization that is responsible for the implementation. It also describes what policy objective the component strives for. The objective may be to improve gender equality, reach an environmental goal or strengthen human rights (Sida, 2014). In order to communicate and follow up international development cooperation, it is of great importance that Sida produces reliable and descriptive statistics. Sida works daily with collecting and aggregating this information. Sida reports statistics to the Swedish government and parliament and to Openaid.se, which is a web-based information service about Swedish aid. Another important and influential receiver of Sida's statistics is the international Development Assistance Committee (DAC) within the Organization for Economic Co-operation and Development (OECD). The statistical information is used as background material for parliamentary propositions, analyses, reports, evaluations, books, etc. The statistics are often required to conduct brief analyses as well as in-depth studies of development-related topics. It is highly important to ensure that the data is correct and reliable, as it is a foundation for new investments. Therefore, quality controls of the data are necessary. The statistical group at the Unit for Analysis and Coordination at Sida is responsible for the delivery of statistics and also performs quality checks of the data. They collect the data through a system where Sida officers, responsible for processing contributions, register the statistics about their components. In this registration phase, the officers are guided by Sida's statistical handbook. The purpose of the handbook is to improve the statistical quality by simplifying the statistical encoding (Statistical group, 2015).

Problem description

According to Sida's statistical group, the accuracy of Sida's classification of e.g. policy objectives is rather low (2015). There are frequent deviations between the officers' classification of contributions and the instructions in Sida's handbook. This could be caused by careless input, difficulties in using the system or troubles with understanding the statistical coding. Some relations between the statistical codes are strict in character. These are known by Sida's statistical group and can thus be used to control and correct the statistics. An example of this could be that when the Sector code starts with "152", the policy objective "Peace and Security" has to be marked. In 2014, an internal investigation of the accuracy of the statistics was made with help of these strict relations. For example, it was found that 11% of the policy objectives were wrongly classified. Sida's statistical group suspects that there are more deviations to be found, but they do not know how many or how to find them. Sida's Helpdesk for Environment and Climate Change did research on four different environmental policy objectives in 2011. They did a review of 139 on-

going contributions and it was found that only 50- 65% of the policy objectives were correctly classified. They did comparisons of the coding with informative

documents about the contributions; more details about their methods and measurements of accuracy can be found in the report “Environmental statistics at Sida - A review of policy markers on climate change adaptation, climate change mitigation, biodiversity and ecosystem services, and environment”. Worth mentioning is that some of the policy objectives were introduced in 2010 and thus had not been used for a long time when the investigation was done. Sadev carried out another assessment of the Swedish aid in 2010. Sadev was the Swedish Agency for Development Evaluation, active between 2006 and 2012. The assessment was made on 30 contributions from Sida that were financial active during 2008. In the evaluation, several quality criteria were investigated; i.e. the accuracy of statistical codes. It was found that the policy objectives often were overused, which means that the objectives were mentioned as relevant for the purpose of the contribution, while it actually was not. This phenomenon was observed in the other report as well. 23-42% of the components had been identified as incorrect, due to wrongly coded policy objectives (Sadev, 2010, p. 20).

Obviously, it can be stated that there is a need of improvement. An on-going project is to update the planning system. The new system will provide more support to the Sida officers in the registration, for example by more restrictions and hands-on guidance. It aims to be ready for use in the beginning of 2016 and hopefully it can contribute to better quality of the data by preventing errors from happening in the first place. Another development area that has been recognized is to make more vigorous and detailed analysis in the control phase, when the contribution data has been stored in the database. There are annually around 5000 active contributions, which indicates the need of an automated analysis tool that can handle all the data. Maybe Sida's data can be used for finding underlying patterns that cannot be detected by a human eye. Perhaps a development of an advanced data algorithm can be a solution to identify the non-representative contributions. There are methods that can detect anomalous data, which could be a solution to find the incorrect coded data.

Purpose

The overall purpose of this study is to find a way for identifying incorrect data in statistics about contributions. The goal is to build an algorithm that determines if a contribution has a risk of being inaccurately coded, based on anomaly detection methods. It is desirable to investigate if descriptions about the contributions, in the form of plain text, can be useful in this problem. In order to obtain the goal, the following questions will be answered:

What kind of informative statistical data is important to include in the study in order to build an anomaly detector?

Which processes and algorithms are preferable to use for this anomaly detection problem?

2. Anomaly detection

In this chapter, research about anomaly detection is presented. The definition of anomaly detection and its different research areas are explained. The characteristics of an anomaly detection problem, followed by a presentation of different application domains the technique is used in, will be discussed.

Background

Anomalies can be defined as patterns or data points that do not conform to a well- defined notion of normal behaviour. In contrast to *noise removal* and *noise accommodation*, where noise/anomalies are viewed as a hindrance to data analysis, *anomaly detection* enables interesting data analysis based on the identification of the anomalies. However, it is noticeable that solutions for noise removal and noise accommodation often are used for anomaly detection and vice-versa (Chandola et al., 2007). Anomaly detection can be defined as follows:

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behaviour (Chandola et al., 2007, p. 1).

Another definition according to Kumar (1995, p. 6) is:

Anomaly detection attempts to quantify the usual or acceptable behaviour and flags other irregular behaviours as potentially intrusive.

These two definitions describe the method well, both how the problem should be understood, and how to deal with it. This is how anomaly detection should be understood in this thesis. Detecting anomalous behaviour can be viewed as a binary valued classification problem, where features are used to build a classifier that decides if a data point is normal or abnormal (Lane and Brodly, 1997). What type of anomaly detection technique that is appropriate to use, depends on the research area, the characteristics of the problem and the application domain (Chandola et al., 2007).

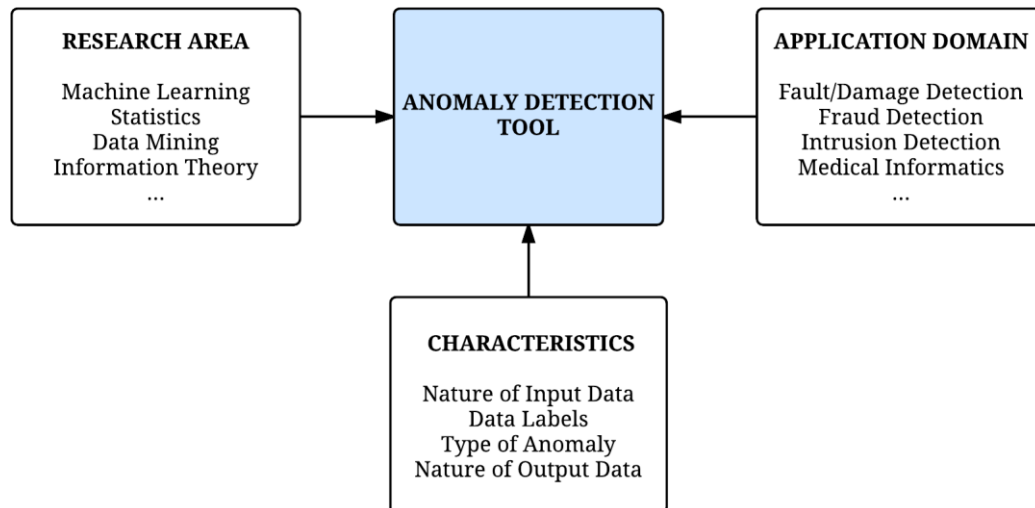


Figure 1 below, explains the relationship between an anomaly detection tool and its key components.

Figure 1. Key components associated with an anomaly detection tool.

(Chandola et al., 2007, p. 4. Treated by authors.)

Researcharea

Detecting anomalies in data has been studied in the field of Statistics since the late 19th century (Chandola et al., 2007). The original outlier detection methods were arbitrary, but in modern times automated and systematic techniques have been developed in many fields, for example in Computer Science (Hodge and Austin, 2004). Many techniques are specifically developed for certain application domains where concepts from diverse disciplines such as Statistics, Machine Learning, Data Mining and Information Theory have been adopted and applied to the specific problem formulation (Chandola et al., 2007). The discipline Machine Learning used in this thesis is described further in the following subchapter.

2.2.1 Machine Learning

Machine Learning is a common approach to the type of problem that is treated in this thesis. The goal is to determine if a data instance is correct or not with help of data- driven prediction. It can also be formulated as optimizing a performance criterion by making predictions from models built on knowledge of past data. Machine Learning is about learning, adapting, recognizing and optimizing – in short, to be intelligent (Alpaydin, 2010). Machine Learning is a scientific discipline that originates from the field of artificial intelligence. Its major purpose is to make a machine learn from data sets and adapt to new

information. Thus Machine Learning uses algorithms that operate by building a model from data inputs and a set of features to make predictions on the output

(Hastie et al., 2009). More ambitious problems can be tackled as more data becomes available. However, the fundamental goal of Machine Learning is to generalize beyond the examples in the available data set. This is because, no matter how much data that is available, it is still unlikely that those exact examples will occur again. Therefore, every learner must embody some knowledge or assumptions beyond the data it's given in order to generalize beyond it (Domingos, 2012). Machine Learning can be applied in supervised, semi-supervised and unsupervised mode; which will be discussed further in 2.3.

Characteristics

The specific characteristics of an anomaly detection problem are determined by several factors. Important aspects for any anomaly detection technique are *the nature of the input and output data, the type of anomaly and the availability of labelled data*. The input data is generally a collection of data instances and each data instance can be described by a set of attributes. The output data defines how the anomaly should be reported, by labels such as normal/anomalous or by scores, where each instance assigns an anomaly score. The nature of attributes can be of different types, such as binary, categorical or continuous, and determines the applicability of different anomaly detection techniques (Chandola et al., 2007). There is also a need of assumptions on what makes a data point an anomaly, for example saying anomalies are not concentrated (Steinwart et al., 2005). Anomalies can be classified into three types; point anomalies, contextual anomalies and collective anomalies. An instance is termed as a point anomaly if it can be considered as anomalous with respect to the rest of the data. If a data instance is anomalous just in a specific context, but not otherwise, it is a contextual anomaly. Contextual anomalies have mostly been explored in time-series and spatial data. The third type, the collective anomaly, is a collection of related data instances that is anomalous with respect to the entire data set. The individual data instances may not be anomalies by themselves, but their occurrence together as a collection is anomalous (Chandola et al., 2007). In this thesis point anomalies will be detected, which are the most preeminent type of research on anomaly detection (ibid).

Depending on the extent to which labelled data are available, anomaly detection techniques can operate in the three different modes; supervised, semi-supervised and unsupervised. Often in real life cases there are only unlabelled samples that are available. If labelled instances for both normal and anomalous classes are available, a supervised mode can be applied. A typical approach in this case is to build a predictive model for normal versus anomalous classes and then compare new data instances against the models to determine which class the new instance belongs to. Generally it is

prohibitively expensive and consequently challenging to obtain labelled data that is accurate and representative of all types of behaviours (Chandola et al.,

2007).

Measurement noise can make it more difficult to distinguish normal and anomalous cases. A human expert often does labelling manually, which can contribute to errors in the data. Further, additional hidden attributes that have not been taken into account when modelling can disturb the classification process (Alpaydin, 2010). Typically, obtaining a labelled set of anomalous data instances which cover all possible type of anomalous behaviour, is more difficult than obtaining labels for normal behaviour (Chandola et al., 2007). Techniques that operate in a semi-supervised mode only require labelled instances for the normal class. This makes these techniques sometimes more widely applicable than supervised techniques. The typical approach used in semi-supervised techniques is to build a model for the class corresponding to normal behaviour, and use the model to identify anomalies in test data. The third mode of anomaly detection techniques, the unsupervised approach, is more of a clustering method. It does not require labelled instances at all. The underlying assumption for this mode is that normal instances are far more frequent than anomalies in the test data. If this assumption is not true then such techniques' alarm rates cannot be trusted (ibid).

This topic will be considered further in chapter 3.2, where the problem solving approach used in this study, together with challenges in this type of classification problem will be discussed.

Application domain

Anomaly detection has been researched within different domains for many years, ranging from fraud and intrusion detection to medical diagnostics. Kumar et al. (2005) highlight the importance of detecting anomalies. Even though outliers are by definition infrequent, their importance is still high compared to other normal events. Take the example of a medical databases and the task of classifying the pixels in mammogram images. Then the abnormal, cancerous pixels represent only a small fraction of the entire image, but are highly important to identify. By detecting these types of anomalies, deviations can be identified before they escalate with potentially catastrophic consequences.. These anomalies may indicate fraudulent cases or they may just denote an error by the entry clerk or a misinterpretation of a missing value code. Either way, detection of the anomaly is vital for database consistency and integrity (Hodge and Austin, 2004). This thesis aims to contribute by applying anomaly detection and text categorization methods in the Aid community. The anomalous cases here are not as alarming as the cases of cancer and fraud, but the misclassified statistical codes are important to identify for Sida to produce reliable statistics.

3. Method

The thesis work has included a literature study about Anomaly detection and Machine Learning methods, to obtain background knowledge of the area. The main work has been of a more practical character. The focus has been the construction of an algorithm for analysing Sida's data and flagging the suspected cases of incorrect coding of the statistics. This was a comprehensive work and will be described in detail in chapter 4, *The data analysis process*. The following three subchapters describe the overall method for this thesis: the software development method, the problem solving approach and the choice of a software programme.

Software developmentmethod

An agile development process has been used in the thesis to make the process flexible, iterative and adaptable to changes. The agile methodologies include the following four terms, which are developed from the Agile manifesto (agilemanifesto.org):

- Iterative*, flexible to requirement updates and changes in functionality
- Incremental*, developing subsystems that allows adding requirements
- Self-organizing*, the team can organize itself to best complete the work
- Emergent*, it is a learning experience, every project is unique

These four key approaches have been taken into account during the study. Additionally, some tools from the agile methodologies have been used in order to maintain high quality of the software development process. For example the *Kanban board* was used to structure the working process. The idea of Kanban is to iteratively split the work into pieces, writing the different tasks on post-it pads and then keep track of them through three different stages. The three stages are: (1) *The backlog*, which is a to-do list of tasks that are planned to be solved in the near future; (2) *In-progress*, the tasks that are being solved at the moment, and (3) *Done*, the tasks that have been accomplished (Saddington, 2013; Olausson et al., 2013). Pair programming is a part of the agile method *Extreme Programming* (XP), which means that two programmers work in front of the same computer. The idea is that one of the programmers writes the code while the other gives comments, recommendations and ask questions, and that the roles switch when needed. With this method better flow and code can be obtained and both programmers can feel a collective ownership over the code (Vliet, 2008). This method has been used in the most important steps of the development of the algorithm, i.e. when working with selection and reduction of features and tuning of modelling parameters. Then one of the programmers could gather information about different programming procedures and give the

programmer recommendations how to apply these. During the work there were frequent meetings with the Statistical group at Sida, especially early when it

was of great importance to get a good understanding of the statistics and the problems with their quality. A brainstorming session with post-its was performed with three out of seven people from the Statistical group (2015). Attention was paid to Sida's expectations for the project, their perception of the difficulties with statistics and their view of important features in the statistics. This was done in order to obtain a collective objective of the project and to get information about which features to include in the study. It was also suitable to participate on a two-day course about the planning system that was held in February, where a lot of knowledge about the input data and the processes was obtained. As a last step when the results were obtained, a final discussion with Sida was held to discuss how to implement and use this type of analysis in their future work.

3.1.1 GNU for statistical computing

In data analysis, it is important to choose an appropriate software programme for the analysis. The programme needs to enable a thorough exploration of data and it needs to be trustworthy (Chambers, 2008). The statistical programme *R* has been chosen as software environment for the statistical computing. *R* is known for being a good and powerful tool to analyse big volumes of data. It provides a wide variety of statistical and graphical techniques, where classification is one of them. Therefore it was suitable for this project. *R* is a functional programming language and is available as Free Software under the terms of GNU General Public License (The R Foundation), which was advantageous to Sida. In addition, free software generally implies that there exists a lot of documentation online about the programme, which makes it easy to learn and use.

The pages *stackoverflow.com* and *statmethods.wordpress.com* etc. were frequently visited during the programming phase. Also while programming in *R*, information about different functions contributed by the *CRAN package repository* was useful.

Problem solving approach

There are several possible approaches to determine whether a data point has a high risk of being inaccurate, which were described in chapter 2. In this study, the problem was formulated as a classification task, which implies that a supervised learning approach could be applied. The data instances used in this study were not labelled as normal or abnormal, but instead other values were used as labels. These were the policy objectives, which were used one at a time as the target class. In Sida's database, each policy objective can be stored as (2) Principal objective, (1) Significant object or (0) Not Relevant. With the intention of simplifying the classification algorithm and use a binary classifier that predicts an objective to be True or False, which in this case corresponds to Relevant or Irrelevant, the (1) and (2) were bunched together. Most supervised Machine Learning tasks are designed to perform best when distinguishing between only two classes, but they can also be extended to multiclass cases (Allwein et. al, 2000). For this problem it was decided that a binary representation was enough. The policy objectives together with how many times a specific policy objective occurred as relevant or irrelevant in the used data set are shown in Table 1.

Table 1. The distribution of classes for the policy objectives.

Policy objective	Relevant	Irrelevant
Environment	7949	9883
Gender Equality	11396	6436
Democracy and Human Rights	12858	4974
Peace and Security	3657	14175
Trade Development	1529	16303
Child and Maternal Health	444	17388
Biodiversity	1036	16796
Climate Change Adaptation	1599	16233
Climate Change Mitigation	1329	16503
Desertification	381	17451

The four last policy objectives in the list above were introduced at the Rio Convention 2010 and fall under the DAC definition of "aid to environment". The other policy objectives have been used since the late 90's. As there are ten different policy objectives, ten separate models were created. Thus, to create the target classes, the policy objectives were assigned the labels Relevant or

Irrelevant. They were then defined as the dependent target variable, Y . The other policy objectives were still included in the dataset and kept their original

form, being a part of the independent input variables, !I to !II. See Table 2 for an example. Here the target variable in the first column is the *Policy objective (PO) Environment*. The other four columns representing the features are *Policy objective Gender Equality*, *Programme based approach*, *Aid type* and *Sector*. These statistical codes are described in more detail in chapter 4.1.1. In this example the first row's descriptive features take the values 0, 0, A and 9 and the target variable has the value Relevant. The 0-values indicate that these features are not important for this specific row/component. While the other two features are categorical variables where Aid type A stands for budget support and Sector 9 is the environmental sector. In the second row is the target variable coded as Irrelevant and the descriptive feature PBA is ticked as being used and the aid type D means that this aid is for experts and other technical assistance.

Table 2. Explanation of the supervised model used in this study.

!I	!II	!II	!II	!II
PO Environment	PO Gender	PBA	Aid type	Sector
Relevant	0	0	A	9
Irrelevant	0	1	D	0

This information was used for training the ten classifiers with a set of chosen input variables, which will be described more in detail in chapter 4. The aim was to train the models to predict as correctly as possible if the policy objective should be marked as Relevant or not. If the classifier predicts a component's policy objective differently than how it was stored, the component is misclassified. To interpret this as an anomaly detection problem, the underlying assumption was that a misclassified component was more likely of being inaccurately coded and therefore anomalous.

3.2.1 Problem characteristics and challenges

When training a supervised learning method, the quality of the training data is one of the most important factors in deciding the quality of the machine. Unfortunately, in real world problems, it is normally not easy to obtain high quality training data sets. For complex data it is possible that erroneous training samples are included, causing a decreased performance of the model (Hartono and Hashimoto, 2007). Since labelled data for Machine Learning often is difficult and expensive to obtain, the ability to use unlabelled data holds significant promise in terms of vastly expanding the applicability of learning

methods (Raina et al, 2007). Various experiments are developed with the aim

of solving the difficulties related to the absence of well-labelled high quality training data in real world problems. The papers “Learning from imperfect data”, published by Hartono and Hashimoto (2007) and “Self-taught Learning: Transfer Learning from Unlabelled data” are two examples from this research area. Hartono and Hashimoto (2007) propose a learning method for a neural network ensemble model that can be trained with a data set containing erroneous training samples. Their experiment shows that the proposed model is able to tolerate the existence of erroneous training samples in generating a reliable neural network. Raina et al. (2007) study a novel use of unlabelled data for improving performance on supervised learning tasks and present a new Machine Learning frame-work called “self-taught learning” for using unlabelled data in supervised classification tasks.

Because of the expense of creating well-labelled data set, Sida could not provide this experimental study with a high quality training set. Of course, this influenced the outcome of the study. The training set that was used to train the classifiers contained erroneous data, which implies that the classifiers have learned how to classify components based on both good and bad examples. This is sometimes called *teacher noise* (Alpaydin, 2010) and worsens the phenomena of *overfitting* that is described below. In this thesis, the problem was overlooked with the aim of examining if a data analysis process with text mining could be used at all for detection of anomalous components. In agreement with Sida’s Statistical group (2015) it was decided to try the problem solving approach using the policy objectives as labels, even if the results from the classification task were not expected to be totally reliable.

Another issue with supervised learning to take into account is overfitting. Even though a model seems to perform well on training data, it might reflect the structure of the training data too closely. The results may be less accurate when applied to a real data set. A model overfits the training data when it describes features that arise from noise or variance in the data, rather than the underlying regularities from the data. One way to understand overfitting is by dividing the generalization error (generated by a particular learning algorithm) into bias and variance. The bias tells us how accurate the model is, on average across different possible training sets. While the variance tells us how sensitive the learning algorithm is to small changes in the training set. A high bias and low variance is an indicator that a learning algorithm is prone to overfitting the model. The ideal is to have both low bias and low variance (Sammut and Webb, 2011). This can be compared to dart-throwing, see Figure 2 for a figurative explanation. The “dart-throwing” model is trained in different manners. If the darts vary wildly, the learner is high variance and if they are far from the bullseye, the learner is high bias.

Noise, like training examples labelled with the wrong class, can aggravate overfitting, by making the learner draw an arbitrary frontier to keep those

examples on what it thinks is the right side. Worth mentioning is that severe overfitting can occur even in the absence of noise. It is easy to avoid overfitting by falling into the opposite error of underfitting (Domingos,2012).

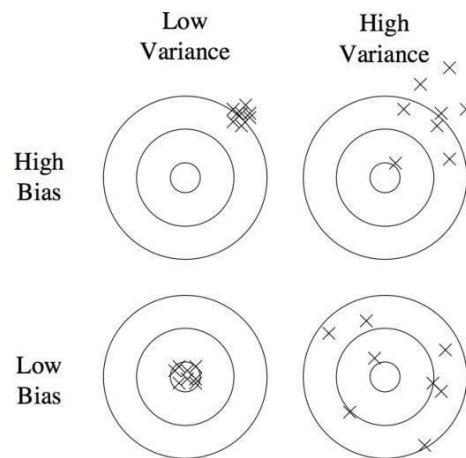


Figure 2. Bias and variance in dart-throwing (Domingos, 2012).

The curse of dimensionality has been another challenging issue to meet in this classification task. The expression was coined by Bellman in 1961 and refers to the fact that many algorithms that work fine in low dimensions, become intractable when the input is high-dimensional. After overfitting, the curse of dimensionality is considered to be the biggest problem in Machine Learning. Naively, one might think that gathering more features never hurts, since at worst they provide no new information about the class. But in fact generalizing correctly actually becomes exponentially harder as the number of feature of the examples grows. This because a fixed-size training set covers a fraction of the input space (Domingos, 2012). Furthermore, the similarity-based reasoning that Machine Learning algorithms depend on breaks down in high dimensions, because the noise from the irrelevant features swamps the predictive variables. To handle this, algorithms for explicitly reducing the dimensionality can be used (ibid). In this study the feature reduction techniques *Mutual Information* and *Principal Component Analysis* were applied to tackle this problem.

4. The data analysisprocess

This chapter is devoted to the procedure that has been carried out with the aim of creating an algorithm that is able to identify inaccurate components in Sida's data. Figure 3 shows the workflow for the data analysis that has been used in this study. As a reader you will find that these main stages also are used as sub headings to guide you through this chapter. This is a theoretical linear model for creating an anomaly detector, but the practical process was far more complex. While processing the data many problems arose along the way. When analysing results from the model generation, issues related to the data preparation occurred, and in order to improve the results further data preparation had to be done. Overall, it was problematic to create a proper data set and there were many turns back and forth, with the goal of obtaining as good results as possible. This iterative approach is one of the core values in agile methods explained in the previous chapter.

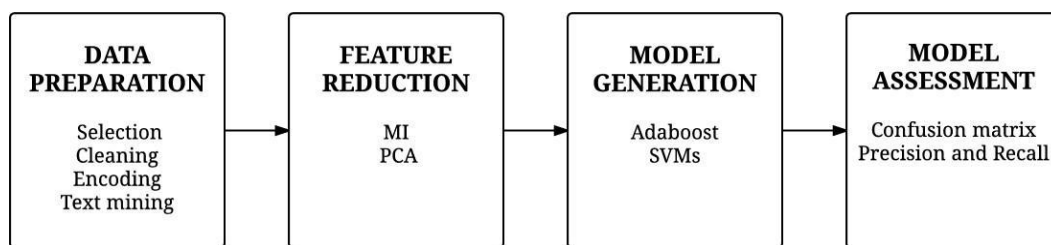


Figure 3. Key steps in the data analysis process. Made by authors.

Datapreparation

The data need to be processed before it is possible to do any analysis. Jonge and van der Loo describes the process as follows: "In practice, a data analyst spends much if not most of his time on preparing the data before doing any statistical operation" (Statistics Netherlands, 2013). Subject to this, the four main step of the preparation phase will now be described more comprehensively.

Data selection

An important step when working with big data sets is the selection of features, which is when you reduce the dimensions and filter out irrelevant features in the data. The selections are made in order to improve the prediction performance, provide faster and more cost-effective predictors and to get a better understanding of the underlying process that generates the data (Guyon and Elisseeff, 2003). After the brainstorming session with the Statistical group (2015), a first selection of which features that should be included in this study

was decided. Some restrictions in what types of features to use were needed, e.g. the receiver country of a contribution, has more than 100 different nominal

values and would have made the data sparse if included. The data that was needed for this project was accessible from Sida's data warehouse and was fetched with an sql query via the RODBC package in R. However, it was a bit tricky to treat duplicates and get an understanding of all different tables in the database, which made this step quite time-consuming. The project abstract field in the database was highly important, due to the aim of using a text mining method for the classification task.

Unfortunately, not all data stored in Sida's database had an abstract. Therefore, a decision was made to only retrieve the contributions that had text in the project abstract column. Sida's database consists of data from both Sida's own contributions and contributions provided by the Swedish Ministry for Foreign Affairs. Only Sida's own contributions were treated in the study. All contributions are defined by their current status; *Rejected*, *Indicative*, *Planned*, *Agreed* and *Completed*. All contributions with status *Rejected* and *Indicative* were excluded in the data set, because they did not include any statistical information that was of interest. As mentioned in the introduction there is a hierarchy in the data where one contribution can contain many components

and all the statistical information is represented on the component level. The descriptive information from the data was interesting for this project, e.g. what kind of organization that implements the contribution, what type of aid is used and within which sector the money aim to support etc. As a lot of information would have been redundant if the contribution ID was used, the component ID was used instead.

When the following selection of data was made, the data set consisted of 17832 rows (components) and 18 columns (the features). The different features are now presented briefly. The ten *policy objectives* that were mentioned in chapter 3.2 were of course included in the feature set. *The project abstract* was used as the feature for representing plain text. The statistical codes *Aid type*, *Sector* and *Implementing organization* were also selected as features. They all are nominal valued features; see Table 3-5 for a presentation of their different denotations.

Data cleaning

Some problems arose while programming due to missing values, which in R are represented by the symbol NA (not available). When examining rows containing NA values, it was discovered that many NA-values appertained to the same components. Based on this knowledge, the rows that included NA-values for any policy objective were omitted first. After this, there were only a handful of NA-values left, which were also removed. This led to a deletion of approximately 7% of the total data set. This is by far the most common

approach to handle missing data and is called “listwise deletion” (Howell, 2012). Although this method results in a substantial decrease in size of the data set, it

does have important advantages. One of them is its simplicity, and another is that you do not get biased parameter estimates in your data set. Alternative approaches are for example *mean substitution* and *regression substitution*. Mean substitution is an old procedure that as the name suggests substituting a mean for the missing data. This is not reliable since the mean value may be far away from the supposed value. A better method is regression substitution that predicts what the missing value should be on the basis of other variables that are present. Depending on which software programme that is used for analysis there also exists several alternative software solutions for missing values. Each program uses its own algorithm for the input data and it can be hard to know exactly which algorithm is used (Howell,2012).

Data encoding

To be able to do the data analysis, the nominal variables need to be transformed into collections of binary variables. These are often called *Dummy variables*, which are variables taking the values of 0 or 1. This representation is useful as it provides more flexibility in selecting modelling methodology (Garavaglia and Sharma, 1998). A transformation of the nominal variable, Implementing organization, was done. It had the values 1-5, representing specific categories. These were transformed to five new binary variables/columns with the values 0 and 1. The same was done for the Sector and Aid type column. It all resulted in a matrix with 50 columns, representing the 18 encoded features. A similar process was also necessary for the text field, but these required a more extensive cleaning and encoding process.

Textmining

Generally, text mining includes two steps, indexing and term weighting. Indexing means that you assign indexing terms for each document, while term weighting is to assign a weight for each term. The weight is a measurement of the importance of each term in a document (Zhang et al., 2011). The data cleaning process was, among others, influenced by Ikonomakis et al. (2005). They illustrate the automated text classification process with Machine Learning techniques in an informative way and many of the steps they presented have been used in this study.

In the data cleaning process the punctuation marks, apostrophes, spaces etc. need to be removed. The purpose of this is to be able to split the words easier and get rid of unimportant characters (Statistics Netherlands, 2013). Stop words were removed to get rid of uninformative words that appear frequently in all documents, such as “the”, “a” and “to”. Sparse terms were also removed to get rid of the least frequent words. Another important pre-processing step for information retrieval is to reduce the amount of inflected words by finding a normalized form of the word. A large number of studies have examined the impact of these transformations of words and many have found an increased retrieval performance when they are used (Hull, 1995). It was applicable in this study, because the actual form of the word was not of interest, but the meaning of the word and the frequency of occurrence of that sort of word were significant.

Stemming and *lemmatization* are common methods for this task, where

stemming removes the suffix of a word and lemmatization produces the basic word form. For example, the words “write”, “wrote” and “written” are stemmed to different words, but the real normalized form is the infinitive of the word, “write”, which is the result of lemmatization (Toman et al., 2006). One of the common approaches for stemming was introduced by Porter in 1980. The strategy he implemented was to compare the end of the words with a list of suffixes. This method worked best in this thesis, as the choices of lemmatization methods in R required comparisons with big dictionaries and it was too computational heavy for this data set. A third common method in text mining is to correct misspelled words, but this was not considered to suit this study since the abstracts contain many abbreviations, for example UN and DAC. These words would have been wrongly corrected and lost their significance if an automated spelling algorithm had been implemented. This problem could have been avoided through inserting such words manually in a dictionary, but this was considered to be too costly for the gain of it. A handful of project abstracts were examined and misspelling did not seem problematic. Besides, people with different nationalities have written the abstracts and the English dialect (American, British etc.) may differ. Some issues according to the dialects were treated by the stemming function, such as words with different spellings as “analyse” and “analyze”.

It was desirable to express the words within the project abstract in a matrix form, the indexed form. In this study a word frequency matrix (WFM) was used. The term frequencies were weighted to enhance an informative WFM. The term frequency- inverse document frequency function (tf-idf) was used, a method that is frequently used in text mining. Featurereduction

After the preparation phase, the number of obtained features needed to be reduced before the modelling could be initialized, this to avoid overfitting that was described in chapter 3.2.1. The goal was to use the minimal number of maximally informative input features. There are numerous methods for finding the most descriptive features. As a result from the literature study and discussions with Michael Ashcroft (2015), two popular approaches were chosen to this project, Mutual Information and Principal Component Analysis.

Principal Component Analysis(PCA)

PCA is a multivariate technique that is useful when there are too many explanatory variables relative to the number of observations and also when the explanatory variables are highly correlated. The aim of PCA is to reduce the dimensionality of a multivariate data set by a transformation to a new set of variables. These are called *the principal components* (PCs) and form a new coordinate system, see Figure 4 for a basic example of how it works. Thus, projecting onto the PCs effects a rotation of the data. The PCs are linear combinations of the original features and are uncorrelated with each other. They are ordered so that the first gives us the direction of maximum variance, and the second has the second biggest variance and is always orthogonal to the first one etc. (Everitt and Hothorn, 2011).

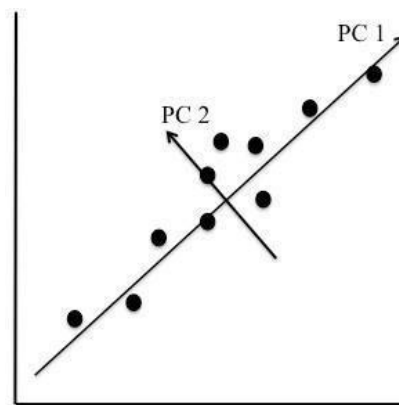


Figure 4. The idea of PCA. Made by authors.

To understand what this kind of projection actually means the *loadings* were examined, which are these linear projections of the data. To reduce dimensionality, a subset of the PCs was chosen by selecting the first ! PCs for modelling. To decide a suitable !, a plotting method called Scree plot was used. The plot shows how the variance changes with the PCs and a threshold could be set where the variance dropped.

After all these processes, the data was presented in a feature matrix, with each policy objective presented as a factor with the values 0 and 1, and all its "best" features with the chosen PCs.

5 Selection of model

When comparing the recall values for the two different classification methods, eight out of ten PO models performed better with Adaboost than SVM in classifying the Relevant class. For the Irrelevant class, four PO models had the exact same recall values, SVM was better in one and then Adaboost performed better on five. According to these results, Adaboost with 40 iterations performed best and was chosen for the final modelling with the test set. Furthermore, Adaboost provides probabilities that were useful for doing further analysis and calculating the anomaly scores.

Moreover, the ten PO models that were learning from the training data performed differently on the validation set. The uneven distribution of relevant and irrelevant tags for some of the POs could be the reason for that as discussed earlier. It could also be that just some patterns are harder to find. The underlying reasons for that could be bad quality of data, unique contributions or the officers might think differently when writing the abstracts and doing the coding. However, a threshold for the recall value was chosen to eliminate some of the models with bad performance. When random guessing if a PO should be marked Relevant or Irrelevant, it would be a 50% chance to guess right, a threshold for the recall performance should then be at least above 50%. Ad hoc it was set to >65% in this project after examining the confusion matrices for each PO model from the Adaboost classification. Four of the models were then removed due to poor performance; these were PO Child and Maternal health, Security and Peace, Desertification and Trade. The other six models were kept and constitute a combined final model that will be presented in the next chapter.

Final model

In this chapter the final model is described in more detail, e.g. with its most important features. The final model is also assessed, mainly in terms of recall. The chapter ends with a discussion about the model's reliability and application.

Importance of features

The boosting algorithm has learnt from the training data that consists of the 30 PC features, where all of them have a relative importance for describing the target variable. This importance measure takes into account the Gini index. This can be displayed in an importance graph, see Figure 13 and 14 that describe the relative importance for the PO Environment model and the PO Climate Change Adaptation model.

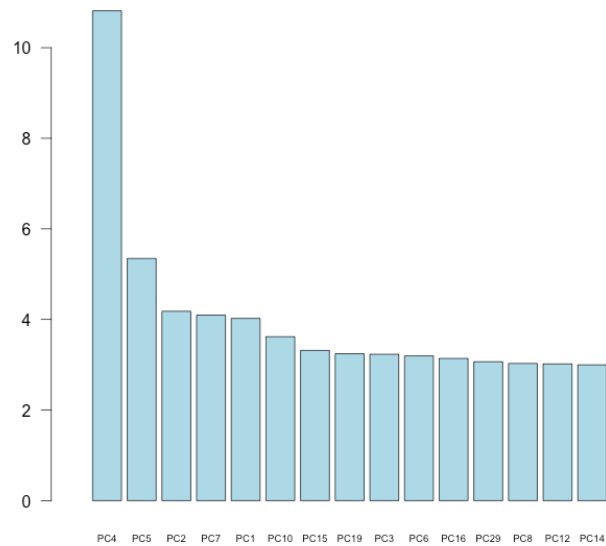


Figure 13. Features relative importance for PO Environment.

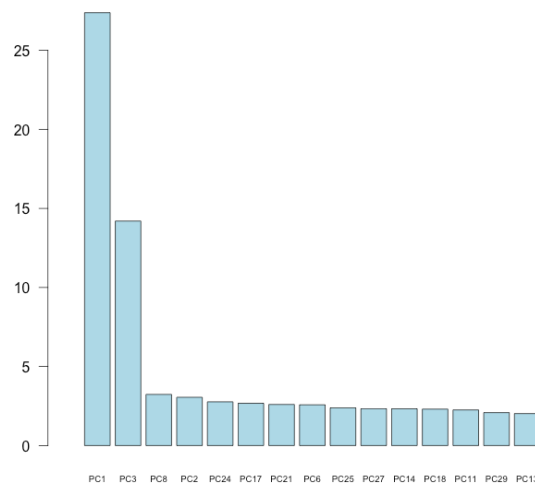


Figure 14. Features relative importance for PO Climate Change Adaptation.

These are the first 15 PC features for both the models. PC 4 and PC 5 are the top two for PO Environment, with PC 4 as the most important one. PC 4 was not included in Table 13 (chapter 5.2) when the different loadings for PC 1 and PC 2 were described, but some interesting features were found when examining the important original features for PC 4. For example the other Rio markers and the stems “environment”, “ecosystem”, “iucn” (International Union for Conservation of Nature) and “solar” influenced PC 4 a lot. PC5 had the stems “scienc” and “univers” as important features. PC 1 is the most important for PO Climate Change Adaptation, followed by PC 3, as you can see in Figure 14. The loadings for PC 1 are presented in Table 14 and show for example that

the other environmental related POs were important. The stems, "forest", "priority" and "expertis", were also significant and were included in PC 3 as well. It

seems reasonable that these features affect the models a great deal, as many of them are related to environmental questions and both the POs target these issues.

The finalassessment

All six confusion matrices for the POs and the precision and recall values are presented in Table 41-46. It can be seen that similar distribution of the results as for the validation set were obtained.

Table 41. Final confusion matrix for PO Environment.

True class	
Predicted	Actual
0	1
0	10
1	10

Table 42. Final confusion matrix for PO Climate Change Adaptation.

True class	
Predicted	Actual
0	1
0	10
1	10

Table 43. Final confusion matrix for PO Gender Equality.

True class		
Predicted	Actual	
0	1	
0	10	10
1	10	10

Table 44. Final confusion matrix for PO Democracy and Human Rights.

True class		
Predicted	Actual	
0	1	
0	10	10
1	10	10

Predicte

d

class

Table 45. Final confusion matrix for PO Biodiversity.

	Relevant	Irrelevant
<i>True class</i>		

Predicted
class

Table 46. Final confusion matrix for PO Climate Change Mitigation.

True class

Predicted
class

Some of the results show that the Relevant class seems to be over used. This can be seen in Table 41, 45 and 46, since those models classify a big proportion of the Relevant labelled components to the Irrelevant class. The phenomenon of POs often being ticked as Relevant, even when they should not was mentioned in the Sadev report (2010, p. 20). This is a positive sign that the models may be useful in their assigned task. Another way of interpreting the results is to take into consideration that those models had less data from the Relevant class to train on, which may be the reason why they classified more components to the Irrelevant class. A similar tendency emerge for the PO models for Gender Equality and Democracy and Human Rights, see Table 43 and 44, where the Irrelevant class instead has a lower recall. To get an overview of the performance of these models, Table 47 shows the final recall values and the total percentage of misclassified components in this final test. In total 1163 components were misclassified, 44 of them were misclassified in more than one model, which results in a total of 1119 unique misclassified components. There were 2138 components evaluated in this test phase, which means that 52% of these components were identified as having a higher risk of being incorrect.

Table 47. Final results for the six PO models.

PO	Recall Relevant	Recall Irrelevant	Total average error
Environment	0.80	0.90	0.15
Climate Change Adaptation	0.79	0.99	0.03
Gender Equality	0.90	0.66	0.19
Democracy and Human Rights	0.95	0.70	0.12

Biodiversity	0.68	0.99	0.03
Climate Change Mitigation	0.74	0.99	0.03

Uncertainties with the classification results

Is a percentage of 52 misclassified components a good result? It is difficult to tell, as there is no right or wrong answer to this question. At a first glance it seems a bit too much. However, as the hypothesis from the Sida's statistical group was that there are many errors in the data than they have found. With their current method, 11% of the contributions were identified as incorrect due to misclassified POs. The other reports about Sida's statistics found around 25-50%, which confirm that this method might give promising results. The other reports identified a general over use of the POs relevance, which the environmental related models showed as well. The difference between the other external investigations though, is that they manually looked at a smaller sample of components, whilst this project lets an automated algorithm do the job and evaluates a big amount of data.

The numbers introduced so far, generated by recall, precision and the confusion matrices, tell us about how the components have been classified. What they do not tell us is anything about the certainty behind these classifications. The certainty of a classifier can actually be measured with help of probabilities. For example, if a component has been classified to the Relevant class with a probability of 0.9, but the component actual class is the Irrelevant class. Our reasoning suggests that the component has a high risk of being an anomaly. Another scenario could be that a component has been classified to the Relevant class with a probability of 0.51 but belongs to the other class. This indicates that the classifier was quite unsure of how to classify the component. Maybe this component should not be considered as anomalous at all? This uncertainty of which class a components belongs to should be taken into account.

Anomalyscore

Adaboost provides probabilities of a component being classified as having a policy objective marked as Relevant or not. The anomaly score was calculated based on these probabilities, where the product of all given probabilities from each classifier for each component were calculated. A weighted anomaly score, *importance score*, was calculated due to the importance of the component's outcome. This task involves four steps that are explained below.

- 1) For each component, and for each model: Calculate the probability that the component was correctly classified. A high probability suggests that the model correctly classified the component.

Therefore, given our reasoning equating misclassification with

anomaly, the probability of a component to be anomalous is:

- 2) Calculate the importance score for an erroneous component.
- 3) The higher the importance score is, the more important it becomes to manually evaluate that component. Therefore a ranking based on the importance score is made as a final step.

Instead of just retrieving the misclassified components as suspected anomalies, a list of the components ranked by their importance score was created. A threshold for the anomaly score was set to >0.7 , to only take the most anomalous cases. This gave 1265 anomalous cases, 65% of the total test set. This threshold is easy to change and provides you with different lengths of lists to check. If it is set to >0.9 , the suspected anomalies are instead 242, and you have instead 11% of a total. However, as this data set consists of components with different Status another selection could be made in this last step.

Sida is most interested in investigating the cases with status Agreed or Planned. Hence, if the threshold with 0.7 is kept, but only these statuses are evaluated, 314 components are found, which is 15% of the test set having a high risk of being anomalous. This is comparable with the 11% Sida found in their investigation in 2014 with the same status condition. This method is flexible and provides a sophisticated way of selecting the components to check. It is a worthy approach since probabilities are more interesting than rough classifications. The final table of the components and their related information is not displayed in this thesis due to confidentiality agreements made with Sida, but in the list below you find the information that is shared with Sida for further investigation.

Evaluation and application

The ranked list of flagged anomalies was then ready for manually investigation. This task was not included in this study, as it requires a lot of expertise, time and money. The Statistical group performed a brief investigation of the top ten most anomalous components from the test set to verify the performance of the classifier. When examining these, Sida officers seemed to have coded five out of ten components wrongly. The number of cases examined was too small for this result to be statistically significant, but it was good that the model at least found some erroneous components.

The hope was of course that all these ten components had been wrongly coded, since that would be a strong indicator that the model actually works. The ambiguous result may be because the model performed poorly. The performance of the model could be improved with (i) better classifiers; (ii) better labelled data; or (iii) alternative anomaly detection methods. Better classifiers may be obtained by using even more substantial methods for settings of modelling parameters and analyses of training errors etc. Higher quality of the training data could also generate better classifiers. Better labelled data would involve a higher quality of the data which likely should improve the result for this data analysis process to a great extent. In practice it would be expensive for Sida to obtain better labelled data, since all components then need to be investigated, eventually corrected and labelled manually. However, if Sida wants to proceed with this kind of quality assurance technique it is to recommend. There are also alternative anomaly detection methods that can be used, for example unsupervised methods, see chapter 6.1 for a brief discussion about this.

To investigate the quality of the model further, a significantly bigger test sample has to be analysed manually. Nevertheless, this algorithm can ease the work for the staff at Sida, not having to look at all components to validate their accuracy, since they receive a set of components that are suspected to be incorrect. This method can provide a set of more probably anomalous components for human inspection. It is more efficient for Sida to get a set of e.g. 100 important cases that have a higher probability of being wrong, than just randomly select components to investigate. Time will tell if the delivered code-scripts will be relevant for their daily operations. Anyhow, Sida believes that this type of data analysis process has potential to increase the quality of aid statistics around the world. They will therefore share this paper to other statistical groups in donor countries.

5. Conclusion

The overall purpose of this study was to find a way for identifying incorrect data in statistics about contributions. This was done within the area of Anomaly detection and Machine Learning. A thorough data analysis process was needed in order to train a model to find hidden patterns in the data. Descriptive features, containing important information about the components were successfully selected and used for this task. The project abstracts were used to retrieve keywords that were informative and helpful in characterizing the type of component. Before the model could be built, there was a need to reduce the number of features even more. This was done with the commonly used methods Mutual Information and Principal Component Analysis, which helped create fewer features containing maximal information. The obtained features after the reduction seemed valid, since relevant principal components influenced the classifiers.

The classification method that performed best was the boosting algorithm Adaboost. It was preferable to use for this anomaly detection problem as it achieved good results concerning accuracy in classifying the policy objective as Relevant or Irrelevant. Six out of ten PO models were chosen due to good recall values. These six classification models were combined to one final classifier, taking the results from each of them into account when calculating the final anomaly score. The product of the probability of a component being classified in the wrong class for each PO model was calculated. The higher the score, the higher the risk of being anomalous. Moreover, the outcome of the contributions was important in order to be able to rank them, so that the most crucial contribution can be checked first. It is impossible to say due to the scope of this thesis, if the identified high-risk components actually are anomalies or not. There are no “true cases” to compare with, but that is normally the case of anomaly detection problems since a thorough evaluation often is too costly. However, an examination of the top ten most suspected anomalies showed that half of them were actually incorrectly coded.

Finally, the idea of using this type of data analysis process can be applied for quality assurance in organizations in the Aid community. It is efficient since it can provide a set of more probably anomalous components for human inspection. The above part is our main code of our project.

```
%% Initialization  
clear ; close all; clc
```

```
%% Part 1: Load Example Dataset  
% We start this exercise by using a small dataset that is easy to  
% visualize.  
%% Our example case consists of 2 network server statistics across
```

%% several machines: the latency and throughput of each machine.

%% This exercise will help us find possibly faulty (or very fast) machines.

```
fprintf('Visualizing example dataset for outlier detection.\n\n');
```

```
% The following command loads the dataset. You should now have the  
% variables X, Xval, yval in your environment  
load('ex8data1.mat');
```

```
% Visualize the example dataset  
plot(X(:, 1), X(:, 2), 'bx');  
axis([0 30 0 30]);  
xlabel('Latency (ms)');  
ylabel('Throughput (mb/s)');
```

```
fprintf('Program paused. Press enter to continue.\n');  
pause
```

%% Part 2: Estimate the dataset statistics

```
% For this exercise, we assume a Gaussian distribution for the dataset.  
%
```

```
% We first estimate the parameters of our assumed Gaussian distribution,  
% then compute the probabilities for each of the points and then visualize  
% both the overall distribution and where each of the points falls in  
% terms of that distribution.  
%
```

```
fprintf('Visualizing Gaussian fit.\n\n');
```

```
% Estimate mu and sigma2  
[mu sigma2] = estimateGaussian(X);
```

```
% Returns the density of the multivariate normal at each data point (row)  
% of X  
p = multivariateGaussian(X, mu, sigma2);
```

```
% Visualize the fit  
visualizeFit(X, mu, sigma2);  
xlabel('Latency (ms)');  
ylabel('Throughput (mb/s)');
```

```
fprintf('Program paused. Press enter to continue.\n');  
pause;
```

%% Part 3: Find Outliers

% Now you will find a good epsilon threshold using a cross-validation set

% probabilities given the estimated Gaussian distribution

%

pval = multivariateGaussian(Xval, mu, sigma2);

[epsilon F1] = selectThreshold(yval, pval);

fprintf('Best epsilon found using cross-validation: %e\n', epsilon);

fprintf('Best F1 on Cross Validation Set: %f\n', F1);

fprintf(' (you should see a value epsilon of about 8.99e-05)\n');

fprintf(' (you should see a Best F1 value of 0.875000)\n\n');

% Find the outliers in the training set and plot the

outliers = find(p < epsilon);

% Draw a red circle around those outliers

hold on

plot(X(outliers, 1), X(outliers, 2), 'ro', 'LineWidth', 2, 'MarkerSize', 10);

hold off

fprintf('Program paused. Press enter to continue.\n');

pause;

%% Part 4: Multidimensional Outliers

% We will now use the code from the previous part and apply it to a

% harder problem in which more features describe each datapoint and only

% some features indicate whether a point is an outlier.

%

% Loads the second dataset. You should now have the

% variables X, Xval, yval in your environment

load('ex8data2.mat');

% Apply the same steps to the larger dataset

[mu sigma2] = estimateGaussian(X);

% Training set

p = multivariateGaussian(X, mu, sigma2);

% Cross-validation set

pval = multivariateGaussian(Xval, mu, sigma2);

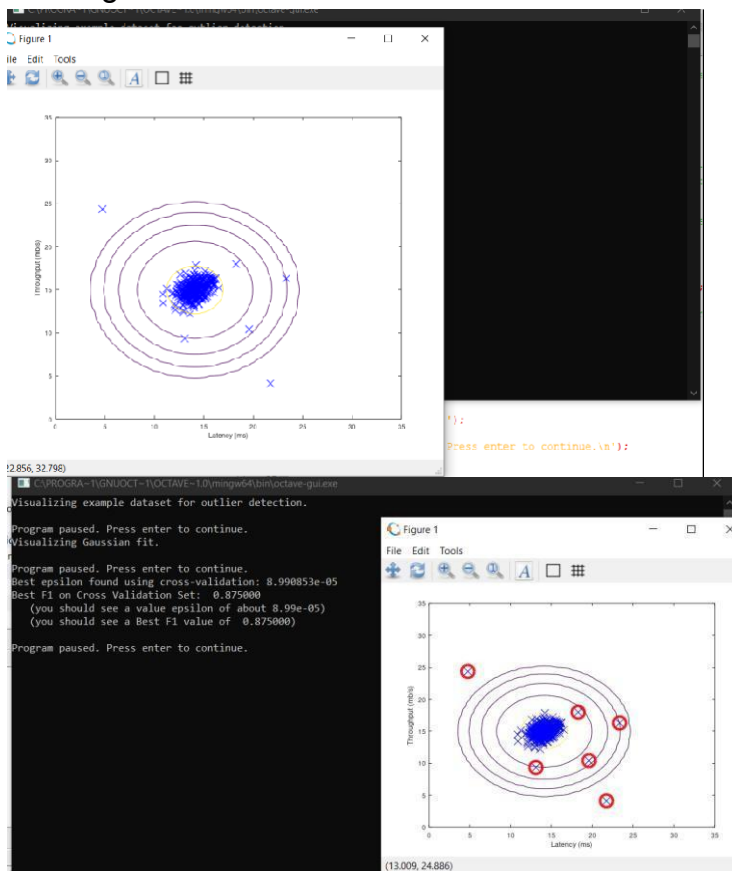
% Find the best threshold

```
[epsilon F1] = selectThreshold(yval, pval);
```

```
fprintf('Best epsilon found using cross-validation: %e\n', epsilon);
fprintf('Best F1 on Cross Validation Set: %f\n', F1);
fprintf(' (you should see a value epsilon of about 1.38e-18)\n');
fprintf(' (you should see a Best F1 value of 0.615385)\n');
fprintf('# Outliers found: %d\n\n', sum(p < epsilon));
```

It would be interesting to find out how other Machine Learning methods would perform on the same task, especially unsupervised learning methods, e.g. Autoencoders. An evaluation of other types of feature selection and reduction methods could also be interesting. For example there are reviews that use Latent Semantic Analysis for text mining, instead of the tf-idf function and PCA.

Concerning text mining methods, all Sida's contributions are also connected to several other documents, for example longer project descriptions and agreements that are stored in Sida's data warehouse. A more extensive research on topic classification and other type of natural languages process with these documents in focus would be interesting to investigate, to find out if similar, or even better results can be obtained. The resultant is above code is given below as screenshots.



References

- ALLWEIN, E.L., SCHAPIRE, R. and SINGER, Y. (2000) Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers, *Journal of Machine Learning Research* 1 (12/00), pp. 113-141
- ALPAYDIN, E. (2010) *Introduction to Machine Learning*, 2nd edition. London: Cambridge Massachusetts, The MIT Press.
- ATKINSON, M. (2013) *Wiley Series on Parallel and Distributed Computing: Data Bonanza : Improving Knowledge Discovery in Science, Engineering, and Business*. New Jersey: John Wiley & Sons, Inc., Hoboken.
- BELLMAN, R.E. (1961) *Adaptive control processes: a guided tour*, Princeton Univ. Press, Princeton.
- BENNETT, K. and CAMPBELL, C. (2000) Support vector machines: hype or hallelujah?, *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 1-13.
- BREIMAN, L. (1994). Bagging Predictors. University of California at Berkeley. No. 421, Berkeley California.
- COVER, T.M. (1965) Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition, *IEEE Transactions on Electronic Computers*, Vol. EC-14, no. 3, pp. 326-334.
- CHAMBERS, J.M. (2008) *Software for data analysis: programming with R*, Springer, New York.
- CHANDOLA, V., BANERJEE, A. & KUMAR, V. (2009) Anomaly detection: A survey", *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1-58.
- CORTES, C. & VAPNIK, V. (1995) Support-vector networks, *Machine Learning*, vol. 20, no. 3, pp. 273-297.
- DAVIS, J. and GOADRIC, M. (2006) The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd international conference on Machine Learning*. ACM.
- DOMINGOS, P. (2012) *A Few Useful Things to Know about Machine Learning*, ACM, New York.
- EVERITT, B. & HOTHORN, T (2011) *Introduction to Applied Multivariate Analysis with R*, Springer New York.
- FREUND, Y. and SCHAPIRE, R. (1999) A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*. 14 (5) (September), pp. 771-780

GARAVAGLIA, S. and SHARMA A. (1998) A smart guide to Dummy variables: Four applications and a macro. In: *Proceedings of the Northeast SAS Users Group Conference*.

GARETH et al. (2013) *An Introduction to Statistical Learning, with Applications in R*, Springer

GUYON, I. and ELISSEEFF, A. (2003) An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3 (March), pp. 1157-1182