Technische Universität München
Faculty of Civil, Geo and Environmental Engineering
Department of Cartography
Prof. Dr. -Ing. Liqiu Meng

Increasing the Statistical Significance for MODIS Active Fire Hotspots in Portugal Using One-Class Support Vector Machines

Vikram Devi Eswaramoorthy

Master's Thesis

Study Course:	Cartography M.Sc.
Supervisors:	Dr.rer.nat. Christian Strobl
	Juliane Cron M.Sc.
Cooperation:	German Aerospace Center (DLR)
Submitted:	June 19, 2017

DECLARATION OF AUTHORSHIP

I hereby declare that the submitted master thesis entitled **Increasing the Statistical Significance for MODIS Active Fire Hotspots in Portugal Using One-Class Support Vector Machines** is my own work and that, to the best of my knowledge, it contains no material previously published, or substantially overlapping with material submitted for the award of any other degree at any institution, except where acknowledgement is made in the text.

Munich, 19th of June 2017 Vikram Devi Eswaramoorthy

ABSTRACT

Forest fires are an important part of the environment as they cause ecological and economical damage, apart from claiming numerous lives. Support Vector Machines (SVMs) coupled with satellite images and sensor information have been proven to aid in the prediction of forest fires and the extent of burnt areas. The National Aeronautics and Space Administration's (NASA) moderate resolution imaging spectroradiometer (MODIS) provides an active fire product in the form of spatial hotspots using an enhanced Collection 6 algorithm since 2015, which basically consists of active fires. This study proposes a one-class SVM based approach to identify the hotspots that are more likely to be a part of large forest fires. For this, Hotspots in Portugal in the time period 2000-2016 were analysed. 38 known large forest fire events across Portugal were used to train the SVM model. As expected, it was observed that these forest fires generally occurred in spatio-temporal clusters when compared to the overall hotspot population. The input parameters for the model were based on spatial and temporal clustering behaviour of the hotspots, external factors such as land cover, temperature, elevation etc. and attributes from the Collection 6 data such as latitude, fire detection confidence, fire radiative power and brightness temperatures. Leave-one-out cross-validation and hold-out validation techniques were used to validate the model. As a result, 79.8 % of the overall hotspots from 2000 to 2017 were classified as ”true” forest fires. These ”true” detections were analysed over time. This led to the identification of other forest fire events that were not included in the training or the test dataset for the model. The results also showed that the model was more sensitive to land cover, temperature, FRP and cluster parameters when compared to other parameters. This proves that the model is responsive to active fires occurring on forest areas and also to spatio-temporal clustering. It is assumed that this approach could be successfully adapted to other study areas as there are no study area dependent input parameters used, except for the latitude, which needs to be taken into consideration. The model can be extended by using other input parameters such as humidity, slope etc. In addition, assigning higher weightage to the more influential parameters such as land cover and spatio-temporal clustering could lead to better results.

Keywords: One-Class Support Vector Machines, MODIS Collection 6 Active Fire Product, MOD14, MYD14, Spatio-Temporal Clustering

ACKNOWLEDGEMENT

First of all, I would like to thank my supervisors, Dr. Christian Strobl (German Aerospace Center) and Juliane Cron (TU München), for their continuous assistance and academic supervision over the period of my thesis. This thesis would not have been possible without Christian's valuable time and advice in the field of geostatistics. Thanks to Juliane for the support all throughout the master study programme, it helped the most. Special thanks to Dr. Christian Geiß from DLR (German Aerospace Center). His knowledge in the field of machine learning was very essential in the successful completion of the thesis.

Next, I would like to thank my family in India; my parents for showing a great deal of interest in the thesis, at times, more than me and my sister, for being my personal research guide. Last but not least, I would like to acknowledge my friends from all over the world; friends from high school, bachelor studies, DLR and the cartography master. Thanks for always being there. It was a fun-filled 3-year ride with my Carto classmates. I have made a lot of great memories with them, which I would cherish life long.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ANN	Average Nearest Neighbour
ANND	Average Nearest Neighbour Distance
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
AVHRR	Advanced Very High Resolution Radiometer
BT	Brightness Temperature
C6	Collection 6
CORINE	Coordination of information on the environment
CV	Cross-Validation
DFD	Deutsches Fernerkundungsdatenzentrum / German Remote Sensing Data Center
DLR	Deutsches Zentrum für Luft und Raumfahrt / German Aerospace Center
FIRMS	Fire Information for Resource Management System
FRP	Fire Radiative Power
GDEM	Global Digital Elevation Model
IMF	Institut für Methodik der Fernerkundung / Remote Sensing Technology Institute
IQR	Interquartile Region
JRC	Joint Research Centre
LC	Land Cover
MAD	Mean Absolute Deviation
MARS	Monitoring Agricultural Resources
METI	Ministry of Economy Trade and Industry
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NDVI	Normalized Difference Vegetation Index
NNI	Nearest Neighbour Index
RBF	Radial Basis Function
ST	Spatio-Temporal
SVM	Support Vector Machine
ZKI	Zentrum für Satellitengestützte Kriseninformation / Center for Satellite Based Crisis Information

1 INTRODUCTION

1.1 Forest Fires and the MODIS Fire Detection Service

Forest fires are an important part of the ecosystem and they affect multiple levels of the ecosystem. A significant section of the living population is constantly being influenced by forest fires directly or indirectly all around the world. A large amount of carbon is emitted by these fires, which result in an increase of atmospheric CO_2, and therefore contributes to climate change [14] [34]. Additionally, forest fires affect the vegetation structure and composition and in some cases, are also a cause of landcover change [14] [31] [13].There are numerous cases of loss of human lives and property due to forest fires over the past. Despite their high influence on all parts of the earth system and their potentially hazardous characteristics, there is a continuous lack of understanding of these high temperature events in the context of future effects of global warming and population growth [12]. Even though they have some shortcomings, satellite imagery and related algorithms have proved to be a huge step towards the successful detection of these events [32].

National Aeronautics and Space Administration (NASA) launched the Terra and Aqua satellites in 1999 and 2002, respectively. Moderate Resolution Imaging Spectroradiometer (MODIS) is a key instrument aboard these satellites. The orbits of these satellites are timed such that Terra passes from north to south across the equator in the morning, while Aqua passes south to north over the equator in the afternoon. This enables the satellites to view the entire Earth's surface every 1-2 days, acquiring data in 36 spectral bands, which are groups of wavelengths [8]. Both satellites have a 1km resolution middle and long-wave infrared bands, which are used to detect actively burning fires. The Collection 6 (C6) MODIS active fire product is available on NASA's Fire Information for Resource Management System (FIRMS) since September 2015 [6].The data is available in the form of spatial points, which are the centers of 1km fire pixels burning at the time of satellite overpass. The data product consists of a number of attributes, calculated using the C6 MODIS active fire detection algorithm [21].

The Deutsches Fernerkundungsdatenzentrum / German Remote Sensing Data Center (DFD) is an institute of the Deutsches Zentrum für Luft und Raumfahrt / German Aerospace Center (DLR) with facilities in Oberpfaffenhofen near Munich, Germany. In cooperation with the Institut für

Methodik der Fernerkundung / Remote Sensing Technology Institute (IMF), DFD is responsible for the development and operation of Zentrum für Satellitengestützte Kriseninformation / Center for Satellite Based Crisis Information (ZKI). ZKI provides a 24/7 service for the rapid provision, processing and analysis of satellite imagery during natural disasters, worldwide. The main goal is to provide relief organisations and public authorities with information products in the form of maps, GIS data etc [10]. The ZKI MODIS fire service for Europe is a fire detection service which operates based on the MODIS sensors on board the Aqua-1 and Terra-1 satellites [1]. It applies the MOD14 algorithm proposed by Giglio et. al. on the images acquired by DLR in Germany (Oberpfaffenhofen and Neustrelitz) [20]. A Graphical User Interface is provided online for the users to access the information through interactive maps. In addition, recent satellite overpasses with metadata are also made available for download.

1.2 Machine Learning and Support Vector Machines

The field of study interested in the development of computer algorithms for transforming data into intelligent action is known as machine learning[28]. It basically aids in learning and predicting situations, objects, processes and any dependent entity using data and artifically intelligent algorithms. Today, everybody uses machine learning algorithms daily activities without knowing it, for example in social networking sites, web search engines etc. In Machine Learning, Support Vector Machine (SVM) are classifiers that use hyper planes to establish classification boundaries within the sample space [28]. One-class SVM is an extension to the support vector algorithm, used for novelty detection [38]. In this thesis, one-class SVMs are used to classify the MODIS active fire pixels based on their data attributes. In doing so, the significance of true fire pixels should be improved.

1.3 Motivation and Problem Statement

Fire fighting and relief operations involve massive financial and manpower. Therefore, firefighting resource management is very critical. It is very important that these resources are used in the right direction in order to save lives, property and nature. The MODIS Active Fire data is a rich source of information that can help us understand the spatial and temporal variations of fire activity on a large scale. This data can reaveal fire patterns when analysed over time and combined with clustering behavior and secondary external information such as weather conditions and relief [32] [22]. However, there are always cases where the detected fires may be false alarms. The MODIS C6 algorithm addresses the issue of false alarms to an extent, resulting in higher validation accuracy than the previous collection 5 Algorithm [21]. Integrating these Spatio-Temporal (ST) analyses and influence of external factors with one-class SVMs could lead us to models that could be used for judging the MODIS fire pixels, which is the main motivation of this master's thesis.

1.4 Research Questions and Objectives

In order to reach out to the above mentioned problem, the following research questions have to be answered:

- How do the weather conditions and the elevation above sea level affect forest fire detections ?
- Do the fire detections occur in ST clusters ? If yes, what is the size of these clusters in terms of space and time ?
- How would the above mentioned input parameters affect the results of the one-class SVM model ?

To answer the research questions, the following objectives were set for this master's thesis:

- To analyse the influence of external weather factors and elevation on forest fire detections.
- To analyse the spatial and temporal characteristics of forest fire detections.
- To combine the above parameters with one-class SVMs to enhance the occurrences of forest fire detections.
- To validate the approach by using the SVM model to detect forest fires over the past and verifying through historical research.

1.5 Thesis Structure

The thesis consists of 6 chapters. The first chapter gives a general overview of the topics involved in the thesis, establishes the the research obejectives, explains the thesis structure and describes the study area used in this thesis. In the second chapter, a literature review of related research studies has been presented, followed by a theoretical background of the technical concepts invloved in this thesis. This thesis consists of two main parts in terms of work flow. The first

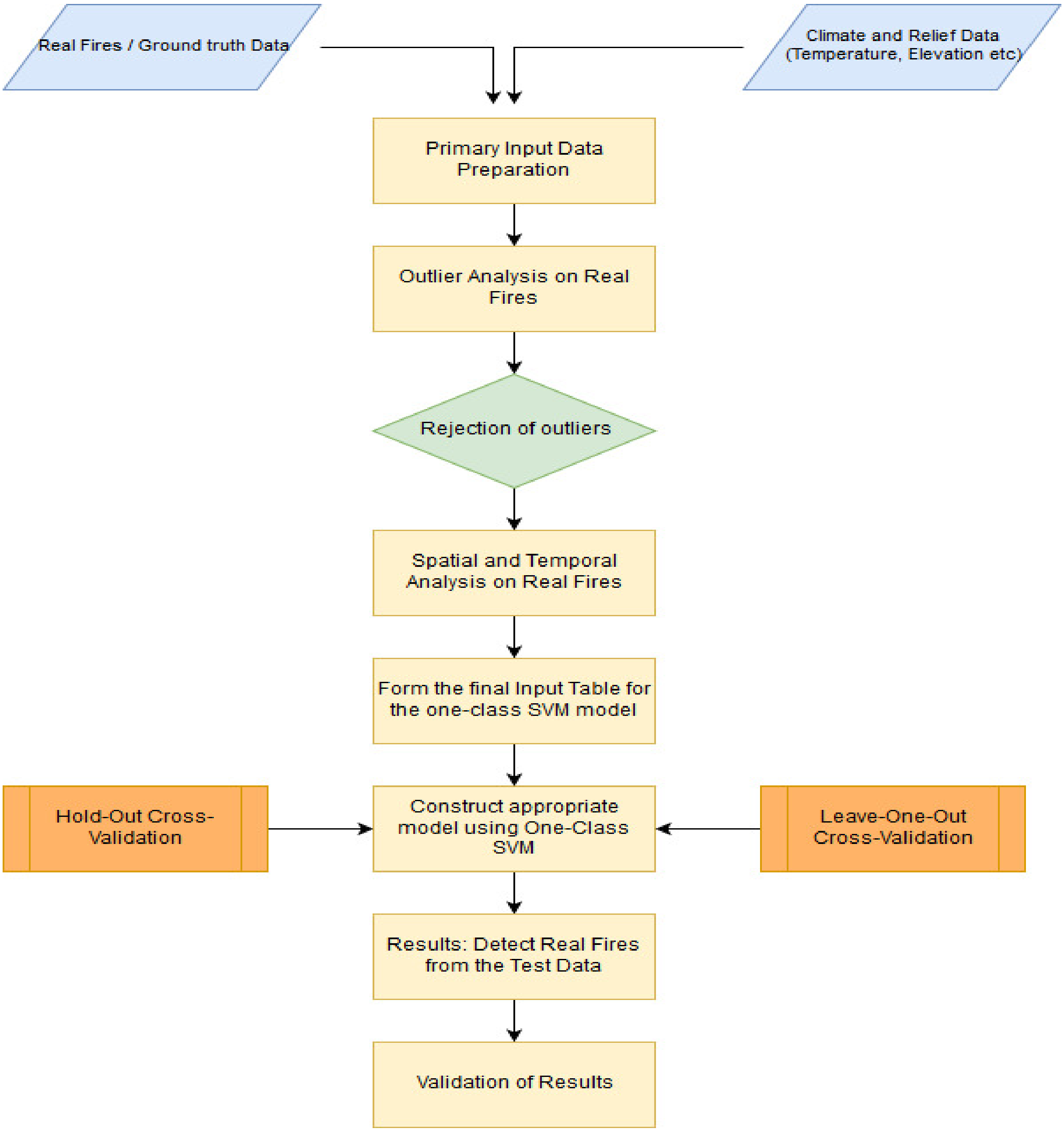

Real Fires / Ground truth Data
Climate and Relief Data (Temperature, Elevation etc)
Primary Input Data Preparation
Outlier Analysis on Real Fires
Rejection of outliers
Spatial and Temporal Analysis on Real Fires
Form the final Input Table for the one-class SVM model
Hold-Out Cross-Validation
Construct appropriate model using One-Class SVM
Leave-One-Out Cross-Validation
Results: Detect Real Fires from the Test Data
Validation of Results

Figure 1: Thesis workflow

part focusses on data collection and preparation (chapter 3) and the second on one-class SVM analysis (chapter 4). Figure 1 shows the workflow involved in the thesis. The first step is data collection (temperature, elevation etc), explained in 3.1. Then, known large fire detections, also referred to as labeled sampels, were digititzed in the sub-chapter 3.2.1. An outlier analysis is performed on these fire detections to filter the data in section 3.2.3. Then, the fire detections are analysed over space and time to get the spatial and temporal parameters in 3.3. In 3.4, the input table is derived which will be the input for the SVM model. Hold-out and leave-one-out Cross-Validation (CV) approaches are used to test the model for over fitting in 4.3. This model is used to detect large fires from the rest of the hotspots in chapter 5, which is basically a set of unlabeled fire detections. Finally, the results are validated by using the model to detect fires over the past and verifying these fires through historical research.

1.6 Study Area

Portugal has a forest cover of around 3.3 million ha, which is more than one third of the total area [11]. Most of the area is composed of woodland, grass shrubs and other light vegetation that is very prone to fire. Due to its Mediterranean climate, Portugal experiences long dry summers. These factors make the forests highly susceptible to wildfires. 2003 and 2005 were two of the worst years in the country's history in terms of forest fire damage [46] [44]. More than 430,000 hectares were burnt in 2003 and around 300,000 in 2005. Portugal is one of Euope's most forest fire affected countries in recent times. The number of fires and the burned area per year has increased significantly in the last decade. Most of the known fires in the time period 2000-2016 were identified in continental Portugal (part of the Iberian peninsula) through historical research. Therefore, continental Portugal was selected as the study area for this thesis (Figure 2). The Azores and Madeira islands were not included in the study area. MODIS active fire data was downloaded for this area in the time period 2000-2016, which is explained in more detail in the next chapter.

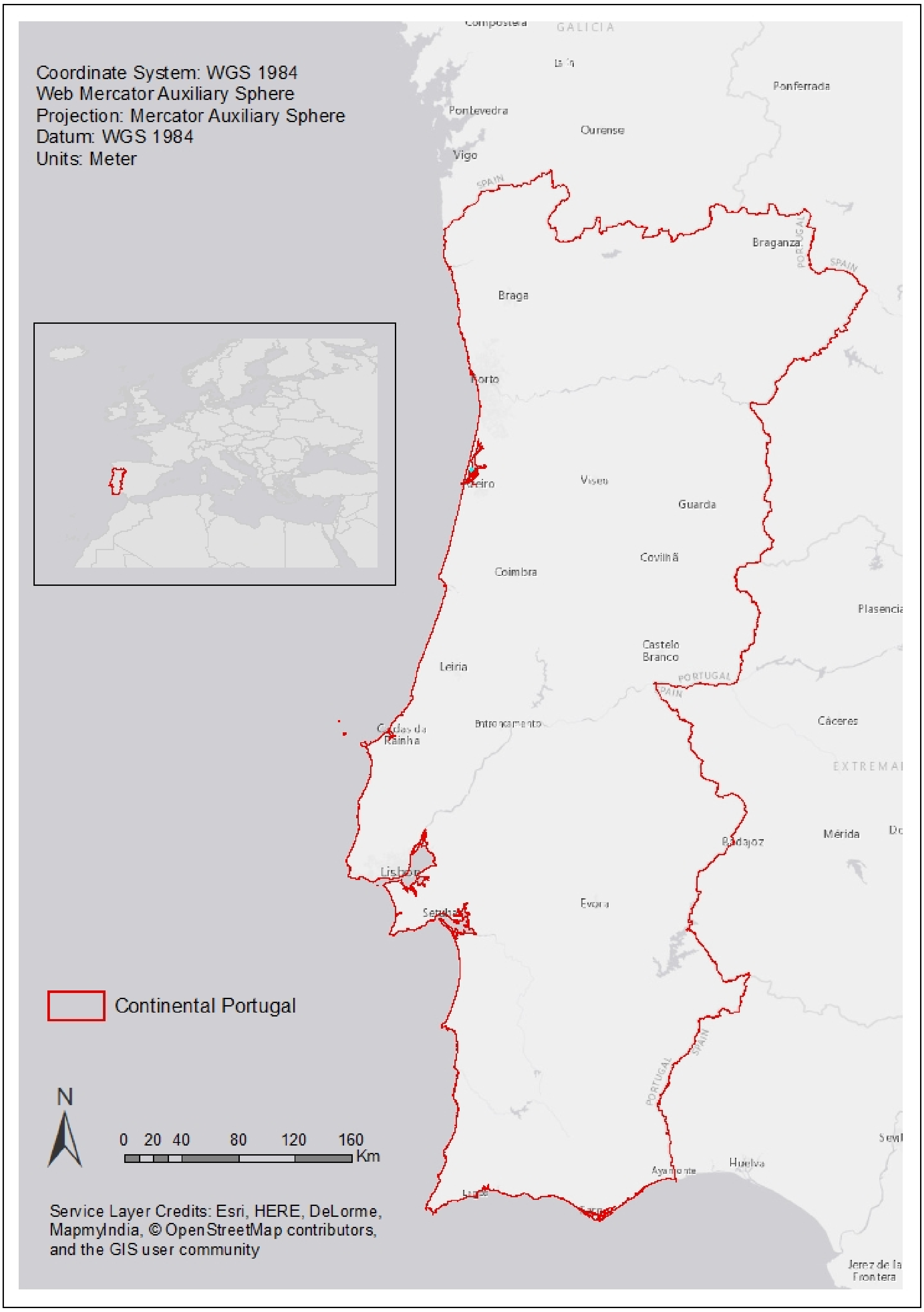


Figure 2: Study area

2 THEORETICAL BACKGROUND

The first sub-chapter of this chapter gives an overview of previous research studies. These studies have described and validated the MODIS C6 algorithm, analysed external factors influencing the MODIS hotspots and used machine learning techniques to predict forest fires. The second sub-chapter describes the geo statistical techniques that are used to analyse spatial point distributions. The third sub-chapter explains the MODIS hotspot data attributes that have been used in this study. The fourth sub-chapter describes the machine learning models and techniques used in this thesis.

2.1 Literature Review

Various research works have been studying MODIS Active Fire Detection algorithms, validating them and analysing the factors that influence them. The relation between Modis hotspots' accuracy and landcover has been discussed in [23] by Stijn et. al. They use Landsat Burned Area maps as reference. The commission errors were generally low and connected to agricultural land cover. The omission errors were high in areas dominated by small burned areas which means that small fires were neglected. In [41], the relationship of the hotspots with burnt area is investigated by Tansey et. al.. It is concluded that validation of the hotspots using global burnt area inventories could lead to better results. Louis et. al. describe the C6 algorithm in detail, presenting it's improved false alarm rate over the Collection 5 alogithm and also the the processiing techniques behind it [21].

It has been proved that forest fires occur in clusters over both space and time by regarding them as ST point patterns and applying statistical techniques [35]. Cluster recognition in ST forest fire squences in Canton Ticino (Switzerland) has revealed clusters over time with varying sizes, which also means that the size of fires depends on external factors and is not constant [33]. Fire data from 1997 to 2003 in Tuscany region, central Italy was investigated for clustering behaviour to conclude that the clustering degree undergoes a significant increment before the largest fire events, so the time period just before a big fire is the most informative for us to define the credibility of the fire [42]. Time series datasets of Normalized Difference Vegetation Index (NDVI) imagery obtained from Advanced Very High Resolution Radiometer (AVHRR) have been used to analyze the fire disturbance (land cover alteration in this case), mainly examining the trend in NDVI values to identify the affected areas in Canada [22]. The results indicate that the fire disturbance was distinct when compared with the unburnt areas, and with temporal variability of NDVI values within the unburnt areas helped to define the recovery times to pre burn levels.

The influence of topographic factors such as elevation and slope have been analysed in North America [36]. The study area was classified as high or low vulnerable zones defined by the month of fire occurence, slope and elevation. Weather conditions (rainfall, remperature, humidity, wind speed), land cover and human activity have been introduced as factors affecting forest fires [30] . A decision making framework is defined that depends on the weather conditions, soil characteristics, land cover and human activity as factors that contribute to forest fire occurrences. On the contrary, studies in the forest regions of Mount Carmel, Israel indicated that the number of forest fires are not correlated to precipitaion in any manner [47]. The increased number of large fires was associated with human activities such as deforestation and also, the location of fires were significantly closer to roadsides.

A Data mining approach was employed in the northeast region of Portugal to predict the burned area of forest fires [16]. Five different data mining approaches, namely multiple regression, decision trees, random forest, neural networ and SVM were tested in this region. The best configuration consisted of an SVM and 4 meteorological inputs (temperature, relative humidity, rain and wind) along with the month of the year and day of the week of fire occurence. Studies in three different regions of Slovenia used other data mining techniques (logistic regression and decision trees) to predict forest fires [40]. The best results in terms of predictive accuracy, precision and kappa statistics were given by bagging of decision trees, which is a meta algorithm to improve the stability and accuracy of machine learning algorithms. It is usually applied to decision tree methods to reduce variance and avoid overfitting.

From all the above mentioned research studies, the following conclusions can be made:

- Forest fire occurences definitely depend on land cover types [23] [30].
- Forest fires occur in clusters over both space and time and the clustering peaks as the fire grows to its full potential[33][35] [42].
- Temperature, wind, rainfall and elevation are the main external factors that influence the

fires[36] [30].

- Machine learning techniques have proven useful to descibe these fires, provided they are used with the right parameters and data attributes [16] [40].

2.2 Spatial and Statistical Analyses

In this sub-chapter, the tools from the field of spatial point pattern analysis that are used to analyse the hotpots (spatial points) are described.

2.2.1 Nearest Neighbour Index

The Nearest Neighbour Index (NNI) is a simple yet effective tool to differentiate a random spatial point distribution from a clustered or an evenly spaced distribution. It compares the Average Nearest Neighbour (ANN) distance of the given distribution to that of a hypothetical random distribution. If the ANN distance is less than that of a random distribution, the distribution under analysis is considered clustered. If the ANN distance is greater, the point features are considered dispersed [27]. It is calculated as shown in equation 1.

$$NNI = \frac{d_0}{d_\mathrm{e}} \tag{1}$$

$$where\ d_\mathrm{e} = \frac{1}{2\sqrt{n/A}} \tag{2}$$

where d_0 is the observed ANN, d_e is the expected ANN for a random distribution, n is the number of point features in the distribution and A is the area under study (usually the bounding box that contains the distribution). Equation 2 is used to calculate d_e. NNI can range from 0, when a distribution is clustered to 2.15, when it is completely regular via 1, when it is random. Figure 3 illustrates these three cases.

However, there are some important points that need to be considered while using this approach. The NNI is very sensitive to the value of A. It is important that the area and the distances have same units (for example, m and m^2). The area can be measured either using a bounding box or a convex hull, but needs to be constant when two distributions are being compared. The sample size should be at least 30 to obtain a meaningful NNI.

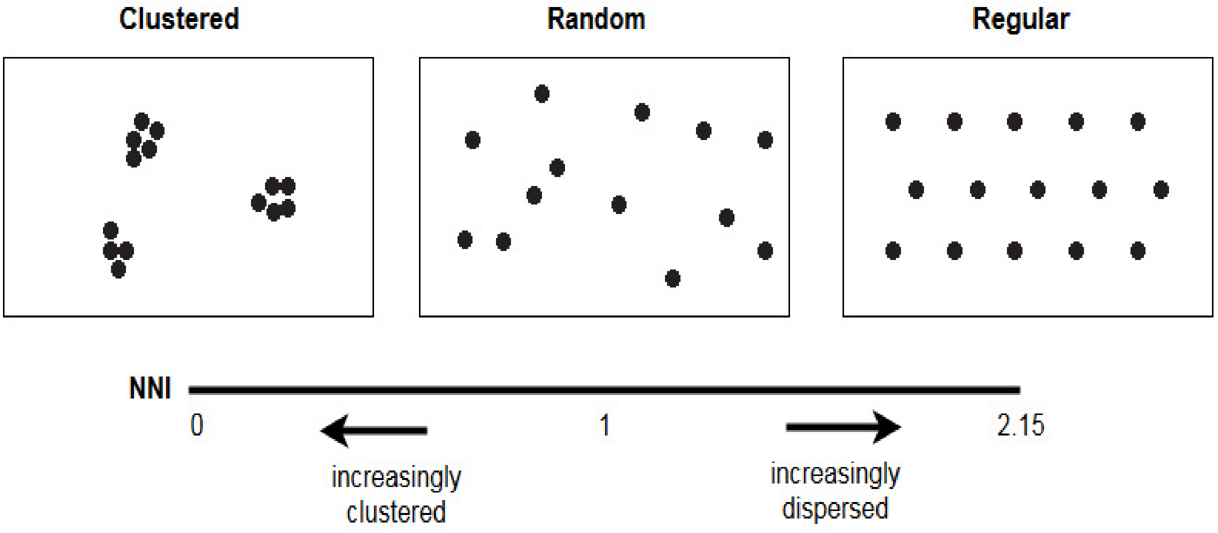


Figure 3: NNI interpretation. Source: [9]

2.2.2 Outlier Analysis

An outlier in statistics can be defined as an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism [24]. The causes of outliers in a spatial point distribution can be due to human errors, noise etc. An inspection of a sample containing outliers would show us characteristic differences within the data in the form of large gaps between the outliers and the "inlying" observations.

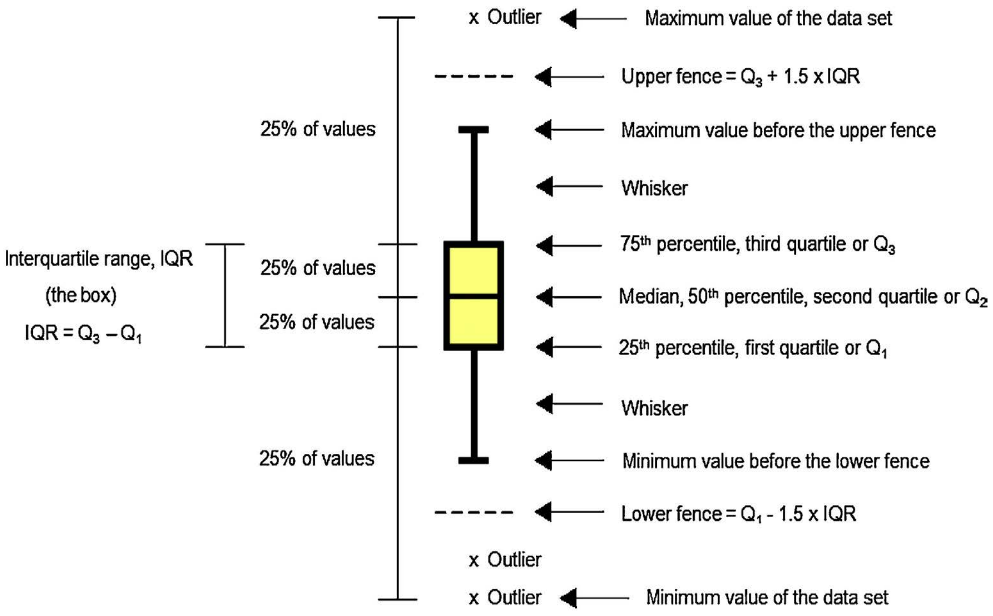

x Outlier
Maximum value of the data set
Upper fence = Q3 + 1.5 x IQR
25% of values
Maximum value before the upper fence
Whisker
Interquartile range, IQR
(the box)
IQR = Q3 – Q1
25% of values
75th percentile, third quartile or Q3
Median, 50th percentile, second quartile or Q2
25% of values
25th percentile, first quartile or Q1
Whisker
25% of values
Minimum value before the lower fence
Lower fence = Q1 - 1.5 x IQR
x Outlier
x Outlier
Minimum value of the data set

Figure 4: Box plot. Source: [17]

The box plot, also known as the box and whiskers plot, is a well known simple display of the five-number summary (minimum, lower quartile, median, upper quartile, maximum), as shown in the Figure 4 [17]. As shown in the figure, the upper fence and the lower fence define the boundaries for a data point to be labeled as an outlier. Any value greater than the upper fence or smaller than the lower fence is labeled as an outlier. The Interquartile Region (IQR) consists of 50 % of the data and the line that splits the IQR is synonymous to the median of the data. The whiskers are extended to the minimum and the maximum values before the lower and the upper fence respectively. The difference between the maximum and the minimum is the range of the data, which can be used to filter the data by removing outliers. This is done in 3.2.3.

2.3 MODIS Collection 6 Active Fire Product

FIRMS's Fire Archive Download Tool provides the MODIS C6 data for this thesis. Data processed in Near Real Time (MCD14DL) is replaced with standard science quality data (MCD14ML) as it becomes available, usually with a 2-3 month lag. Both MCD14DL and MCD14ML are processed using the MOD14/MYD14 (Aqua/Terra) Fire and Thermal Anomalies product. This data can be downloaded as ESRI shapefiles or csv files. This sub-chapter gives an overview of the hotspot attributes that have been used in the study. The detailed description of the C6 algorithm and the following attributes has been explained by Giglio et. al in [21].

2.3.1 Brightness Temperatures

Brightness Temperature (BT) is actually a measure of the photons at a particular wavelength received by the spacecraft [6]. It is presented in units of Temperature (Kelvin). BT is a measure of brightness and not the state of the surface (hot or cold). It should also be noted that the BTs measured by the satellites are generally less than those that are measured at the ground. The C6 algorithm uses the BTs derived from the 4- (channel 21 and 22),11-(channel 31) and 12-micrometer(channel 32) MODIS channels which are denoted by T_4, T_{11} and T_{12} respectively [21].

MODIS has two 4- μm channels, numbered 21 and 22, both of which are used by the algorithm. Channel 21 saturates at nearly 500 K and channel 22 at 331 K. The low saturation channel (22) is less noisy and has a small quantization error. Therefore T_4 is derived from this channel whenever possible. However, when channel 22 saturates or has missing data, it is replaced with channel 21 BT. T_{11} is computed from channel 31, which saturates at 400 K for the Terra MODIS and

340 K for the Aqua MODIS. T_{12} is derived from Channel 32 and used for cloud masking [20]. For daytime observations, 0.65-, 0.86- and 2.1-micrometer (Channels 1,2 and 7) reflectances are used to reject false alarms and cloud masking. Table 1 gives an overview of the channels used in the algorithm.

Channel number	Central wavelength (μm)	Purpose
1	0.65	Sunglint and coastal false alarm rejection; cloud masking.
2	0.86	Brightsurface, sun glint, and coastal false alarm rejection; cloud masking.
7	2.1	Sunglint and coastal false alarm rejection.
21	4.0	High-range channel for active fire detection.
22	4.0	Low-rangechannel for active fire detection.
31	11.0	Activefire detection, cloud masking.
32	12.0	Cloud masking.

Table 1: MODIS channels used in the detection algorithm. Source : [21]

T_4 and T_{11} are used to identify potential fire pixels and are included as attributes with names “BRIGHTNESS” and “Bright_T31” in the hotspot data.

2.3.2 Confidence

The detection confidence estimates range between 0 and 100%. It intends to help users judge the fire pixels in terms of quality. It is used to assign one of the three classes (low-confidence fire, nominal-confidence fire or high-confidence fire) to all fire pixels within the fire mask. It is calculated as the geometric mean of five sub-confidence parameters which are defined in terms of T_4, the number of adjacent water pixels (N_{aw}), the number of adjacent cloud pixels (N_{ac}), the standardized variables z_4 and $z_{\Delta T}$. The ramp function S(x;α,β) is used to derive the sub-confidence parameters. It is defined in the equation 3.

$$S(x;\alpha,\beta) = \begin{cases} 0; & x \leq \alpha \\ (x-\alpha)/(\beta-\alpha); & \alpha < x < \beta \\ 1; & x \geq \beta \end{cases} \tag{3}$$

$$z_4 = (T_4 - \bar{T_4})/\delta_4 \tag{4}$$

$$z_{\Delta T} = (\Delta T - \bar{\Delta T})/\delta_{\Delta T} \tag{5}$$

Equations 4 and 5 are used to calculate z_4 and $z_{\Delta T}$. Here, $\bar{T_4}$ is the Mean 4-μm BT, δ_4 is the 4-μm BT Mean Absolute Deviation (MAD), $\bar{\Delta T}$ is the Mean BT difference, ΔT is the BT difference (T$_4$ - T$_{11}$) and $\delta_{\Delta T}$ is the BT difference MAD. The sub-confidence parameters are derived using the following equations:

$$C_1 = S(T_4; {T_4}^*, 360\text{K}) \tag{6}$$

$$C_2 = S(z_4; 3.0, 6) \tag{7}$$

$$C_3 = S(z_{\Delta T}; 3.5, 6) \tag{8}$$

$$C_4 = 1 - S(N_{ac}; 0, 4) \tag{9}$$

$$C_5 = 1 - S(N_{aw}; 0, 4) \tag{10}$$

Some adjustments are made to the sub-confidence parameters, for daytime and nighttime fire pixels, on both land and water, which can be found in [21].

2.3.3 Fire Radiative Power

The Fire Radiative Power (FRP) is a measure of the rate of radiant heat output from a fire [37]. The pixel-integrated FRP in MW(megawatts) is an attribute available in the active fire data. In C6, the FRP is approximated as

$$FRP \approx \frac{A_{pix}\sigma}{a\tau_4}(L_4 - \bar{L_4}) \tag{11}$$

where L$_4$ is the 4-μm radiance of the fire pixel, $\bar{L_4}$ is the 4-μm background radiance, A_{pix} is the area of the MODIS pixel, σ is the Stefan-Boltzmann constant (5.6704 $\times$ 10^{-8} W m^{-2} K^{-4}), τ_4 is the atmospheric transmittance of the 4-μm channel and a is a sensor-specific empirical constant [21]. Compared to Collection 5, there was a consistent decrease in FRP for the vast majority of fire pixels, but with occasional slight increases for small fractions of comparatively high intensity fire pixels.

2.4 Machine Learning

SVMs are machine learning based classifiers which are very useful in complex classification problems. In this thesis, one-class SVMs, a special case of SVMs, are used to increase to detect large forest fires.

2.4.1 Support Vector Machines and One-Class Support Vector Machines

SVMs rely on automated learning of known (labelled) data, to model the data's characteristics and behaviour. This input data, which are vectors in the sample space, are linearly or non-linearly projected to a high dimensional feature space [15]. The non-linear projection of vectors is done using kernels, which will be explained later in this chapter (Figure 6). In this feature space, a linear decision surface or a hyperplane is constructed to classify the input vectors. The dimension of the feature space depends on the number of labelled classes in the input data. The input data is usually split into training and test datasets. The training dataset is used to learn the labeled classes. The goal of the SVM is to produce a model that could predict the classes (labels) of the vector instances of the test dataset given only the test data attributes [25]. Figure 5 illustrates the simple case of a two-group classification problem in feature space. Note that the vectors are linearly seperable on the feature space, not the sample space.

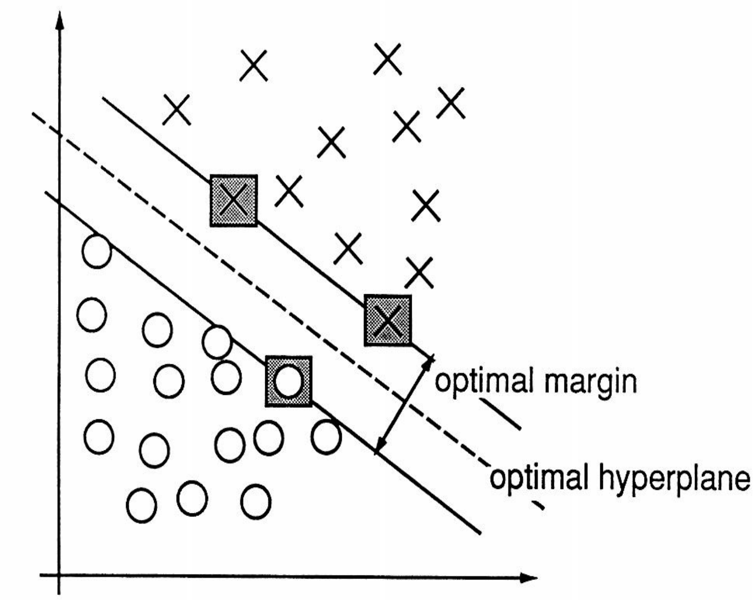


Figure 5: Example of a seperable classification problem in a 2-dimensional feature space. Source: [15]

The grey squares represent the support vectors and the dotted line represents the hyperplane. The support vectors are the outer most vectors of a class which decide the classification boundary.

The optimal hyperplane is defined as a linear decision function with maximal margin between the two classes as shown above.

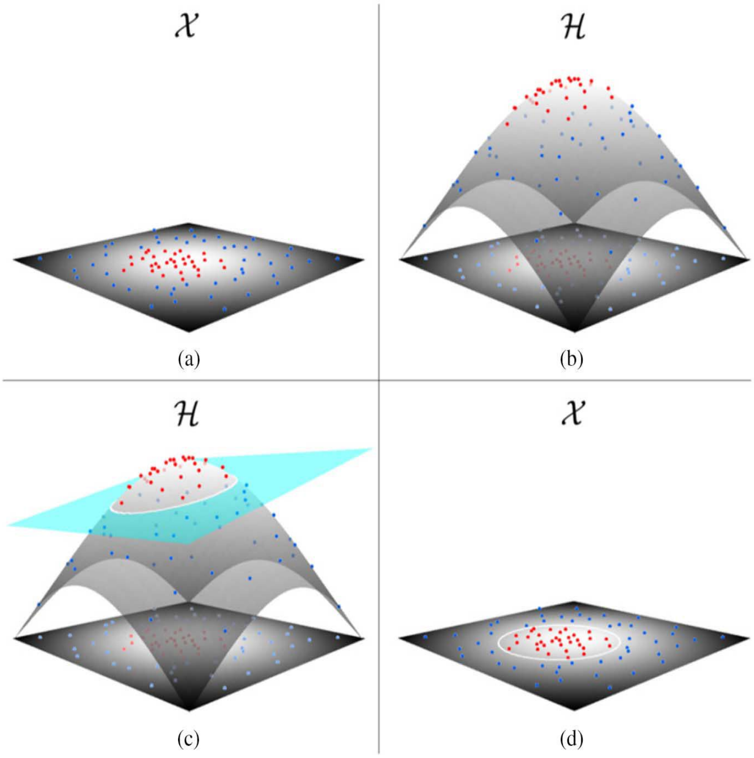


Figure 6: Procedure for generation of a non linear decision function by SVM. Source: [19]

Figure 6 shows the general principle of a non-linear SVM [19]. Dataset with two classes, denoted by red and blue dots, are inseperable in sample space χ [Figure 6(a)]. These dots (vectors) are mapped to a higher dimensionality, where a linear seperation becomes possible with the help of a hyperplane [Figure 6(b,c)]. This corresponds to a nonlinear decision function in χ [Figure 6(d)].

A modification of the maximal margin hyperplane was proposed in [15] where the objective was to find a function *f(x)* that permits errors as long as they are less than ϵ, but will not accept any longer deviation than this. At the same time, the function has to be as flat as possible, since it is a hyperplane. This problem leads to the formulation as shown in equation 12 [15] [39]:

$$\begin{aligned} &minimize\ \frac{1}{2}\left\|w\right\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i{}^*) \\ &subject\ to \begin{cases} y_i - \langle w, x_i \rangle - b & \leq\ \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i & \leq\ \varepsilon + \xi_i{}^* \\ \xi_i, \xi_i{}^* & \geq\ 0 \end{cases} \end{aligned} \tag{12}$$

(x_i, y_i) are the coordinates of the vectors of the labeled data. w represents a vector perpendicular to the hyperplane. ξ_i and $\xi_i{}^*$ are slack variables introduced to cope with the constraints of the optimization problem. The constant C (Cost) is the trade-off between the flatness of *f* and the extent to which the deviations larger than ϵ are tolerated. To extend the SVM to nonlinear functions, a Lagrange function is first constructed from 12. This enables the introduction of kernels, which basically preprocess the input vectors before applying the support vector algorithm. In this thesis, the Radial Basis Function (RBF) kernel is used because it is the most reasonable choice for mapping sample vectors in a non-linear way, which is required for the non-linear data involved in this study. Furthermore, it is also a relatively simple kernel when compared to the polynomial kernel in terms of number of hyperparameters [25]. The mathematical form of an RBF kernel is shown in equation 13

$$K(x_i, x_j) = exp(-\gamma \left\|x_i - x_j\right\|^2), \gamma > 0 \tag{13}$$

γ is a kernel parameter.

A general tutorial to the mathematical concepts behind the support vector algorithm is given by the authors of [39]. The concepts have been explained in detail by Vapnik, who is the co-inventor of the support vector machine method, in 2013 in [43].

One-Class SVM is an extension to the support vector algorithm proposed by Schölkopf et. al. in [38] for the case of unlabelled data where the input dataset has only one class. The algorithm returns a function *f* which is equal to +1 in a small input dataset *S*, by mapping most of the points in *S* and -1 elsewhere. In simple words, any vector in an unlabeled dataset that is similar to the vectors in *S* would produce a value of +1 for the function *f* and any vector that is not, will produce -1. The strategy is mostly similar to SVM : using Lagrange multipliers and a kernel to map the data into a feature space, and seperate them from the origin with maximum margin. For a vector in the unlabeled dataset, the function evaluates the side of the hyperplane that the vector falls on, to determine its class (+1 or -1). To seperate the dataset from the origin, the following quadratic equation is solved :

$$\begin{gathered} minimize\, \frac{1}{2}\left\|w\right\|^2 + \frac{1}{\nu l}\sum \xi_i - \rho \\ subject\,to\, \langle w, \phi(x_i)\rangle \geq \rho - \xi_i,\ \xi_i \geq 0 \end{gathered} \tag{14}$$

Here, ρ is the distance to the origin. The parameter ν ∈(0,1) , also known as nu, is actually a trade off between the upper fraction of training margin errors and the lower fraction of number of support vectors. A large number of support vectors would result in a rigid classification boundary, resulting in more misclassification errors. For example, if the ν is 0.1, it means that atmost 10 % of the training dataset is allowed to be misclassified and at least 10 % of the training dataset need to be included as support vectors for the classification boundary. Having a lower limit of 10 % for the number support vectors might lead the model to use too many support vectors. Therefore, the second condition with a cap on the misclassification error margin prevents the model from using too many support vectors.
In this thesis, one-class SVM is used to learn the labeled dataset containing vectors belonging to large forest fires and then detect similar vectors (fire pixels) from an unlabeled dataset (explained in detail in chapter 4.1).

2.4.2 Overfitting and Underfitting

Overfitting and Underfitting are the two common problems encountered in machine learning caused by the level of model complexity. Overfitting occurs when a model includes idiosyncrasies based on the labeled/input data which might not be reliable generalizations for the purpose of predictions involving other unlabeled data [18]. In short, it is the case when the model is too perfect for the labeled samples resulting in errors in predictions for the unlabled ones. Underfitting is the opposite case, when the model is too simple ie. when too many generalizations are made for the labeled data resulting in both training errors and validation errors. This is illustrated in Figure 7. It can be seen that the underfit case has too many misclassifications caused by a fairly simple classification boundary. The overfit case has no missclassifications in the training data but has a very complex boundary.

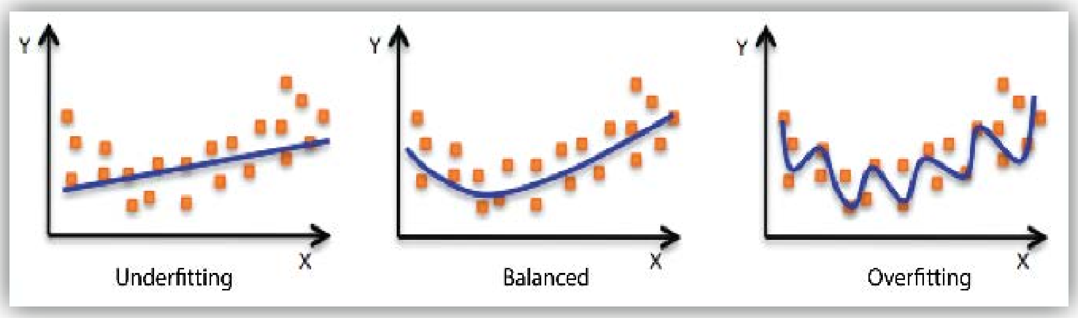


Figure 7: Overfitting and Underfitting. Source: [7]

The optimal fit is achieved with the right level of model complexity. Figure 8 illustrates the three cases graphically as a function of model complexity [29]. Underfitting results in both training and validation errors, as shown in Figure 8. Overfitting is more of a validation problem in terms of error rate.

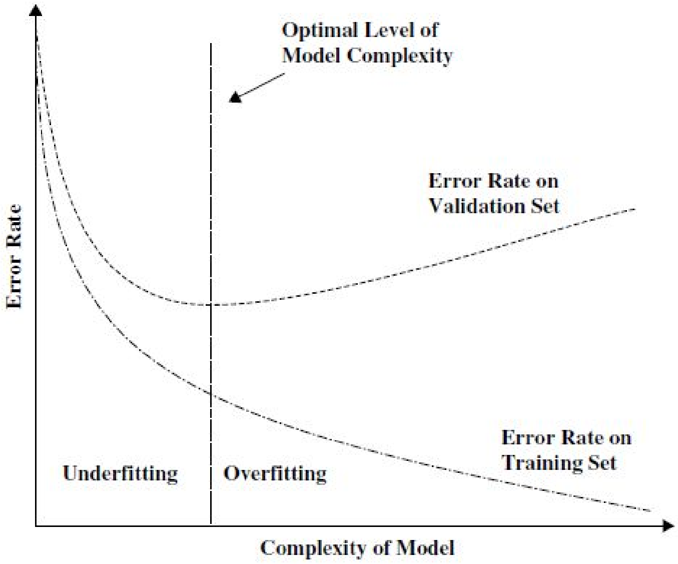


Figure 8: Overfitting and Underfitting in terms of model complexity. Source: [29]

In this thesis, CV techniques are used to prevent over and underfitting which are described in detail in 4.2 and 4.3.

2.4.3 Sensitivity and Specificity

For decision systems like SVM, a false alarm is not as important as a missed correct alarm, which is analogous to the objective of this thesis. The focus is to increase the significance of the true fire pixels, by making sure that no true pixel is classified as false. For this, it is essential to control the trade off between the so called *false positives* and *false negatives* [45]. The confusion matrix shown below categorizes a test point into one of the four categories : True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

		Expected outcome	
		P	**N**
Machine outcome	**P**	True Positive	False Positive
	N	False Negative	True Negative

Sensitivity and specificity are given by the formulae [45]:

$$sensitivity = \frac{TP}{TP + FN} \tag{15}$$

$$specificity = \frac{TN}{TN + FP} \tag{16}$$

In respective to this thesis, sensitivity gives the fraction of correctly classified test data that are true large fires and the specificity gives the fraction of correctly classified non-fire pixels that are not large fires. This analysis is used to estimate the influence of each input parameter on the one-class SVM model.

3 DATA COLLECTION AND PREPARATION

The first sub-chapter describes the processes involved for collection of different input parameters and data required for the one-class SVM model. The second sub-chapter explains the techniques used to collect and filter the data for known forest fires (labeled data). The processes involved in computing the ST input parameters for the model are introduced in the third sub-chapter. These input parameters are used by the one-class SVM model in the next chapter to detect large forest fires.

R programming language was used for all the data processing, analysis and modelling tasks in this thesis. The R packages that were used in the thesis are given below along with the load statements:

```r
library(raster) #for rasterizing images
library(sp) #for using spatial objects: fire pixels as points
library(stats) #for plotting the fires over time
library(rgeos) #for Nearest neighbour interpolation
library(rgdal) #reading and writing shape files
library(car) #for the recode function to repair data
library(plyr) #for data wrangling functions
library(ggplot2) #Plotting
library(dplyr) #for data wrangling functions
library(spatstat) # for point pattern analysis
library(e1071) #for SVM functions
```

The R code included hereon, has to be followed in the given order in combination with the required input data, for successful application of the consecutive steps invloved in this thesis.

3.1 Collection of Input Data Sets and Parameters

3.1.1 MODIS Collection 6 Data

The MODIS C6 data attributes that have been used in this thesis are described in section 2.3. The data was downloaded as an ESRI shapefile for the time period 2000-2016, reason being that the data was made available starting from November, 2000 [6]. As mentioned earlier, the fire locations are the centers of 1 km pixels that are detected by the C6 algorithm as containing one or more fires within the pixel. Figure 9 shows the distribution of fire pixels in Portugal.

Around 53,300 fire pixels were detected in 2000-2016 in Portugal. More than 70 % of these pixels were located in northern and central Portugal. The points shapefile was projected to the coordinate system $WGS\_1984\_UTM\_Zone\_29N$ using QGIS, because of Portugal's location and the unit of measurement is meters which is the standard unit of distance measurement in this thesis. Note that all the shapefiles and rasters used in this thesis are projected to the same

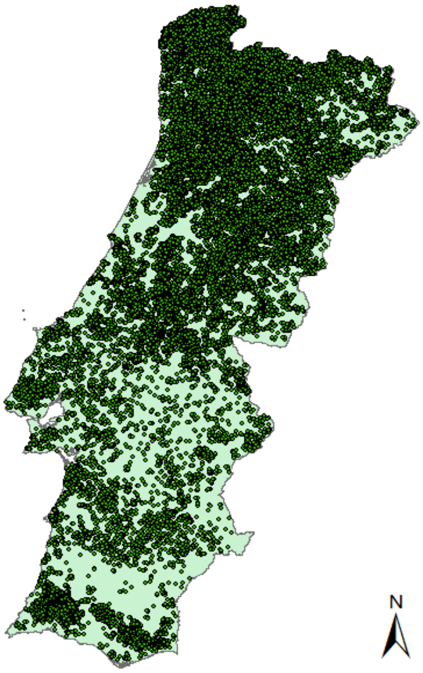


Figure 9: MODIS fire pixels in Portugal in the time period 2000-2016.

coordinate system. FRP, channel 21 and 31 BTs and fire detection confidence are the attributes acquired from this data as input parameters for the one class-SVM model. Because the FRP is an important characteristic of a fire and BTs are an integral part of the algorithm. The confidence gives a measure of the quality of the fire pixels, which might aid in the outlier analysis to remove the unreliable pixels in the sub-chapter 3.2.3.

3.1.2 Land Cover

Since the goal is to highlight the fire pixels that occur in forest areas, Land Cover (LC) data for Potugal is used to create an input parameter for the SVM model. European Union's Coordination of information on the environment (CORINE) LC 2012 data was used. The data is freely available at [2]. The temporal extent of this data is 2011/2012 and it was published in 2016. Figure 10 shows the distribution of land cover in Portugal. The green areas depict different types of vegetation cover, mostly dominated by forests. The extent of the forest cover in Portugal can be seen, which is very high. The LC data was downloaded as raster, with a spatial resolution of 250 meters and 44 standardized LC classes.

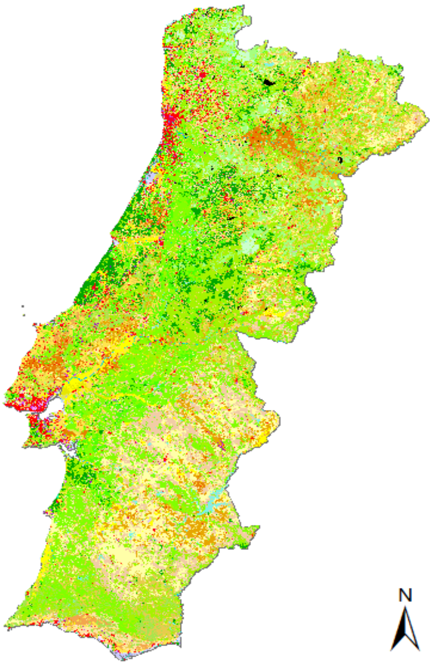


Figure 10: Land cover distribution in Portugal.

The following R code was used to extract the LC classes lying beneath the fire pixels:

```
#the point shapefile
path <- ”Modis Fire Pixels path”
#this object ”shp” will be used throughout the thesis
shp <- readOGR(dsn = path,layer = ”name of shape file”)

#import LC data
PortugalLandCover <- ”LC file path”

#create raster lulc
lulcLayer <- raster(PortugalLandCover)

# Extract the raster values underlying the fire points
shp@data$LULC_val <- extract(lulcLayer, shp)
```

The percentage distribution of the LC classes is given in Appendix A (A.1).

3.1.3 Elevation

Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) Global Digital Elevation Model (GDEM) version 2 data was used to obtain the elevation values for the fire pixels. ASTER GDEM is a product of Japan's Ministry of Economy Trade and Industry (METI) and NASA [26]. The elevation is available in units of vertical meters. Individual granules were downloaded for the area of interest and mosaiced together using QGIS to form the final raster, as shown in Figure 11.

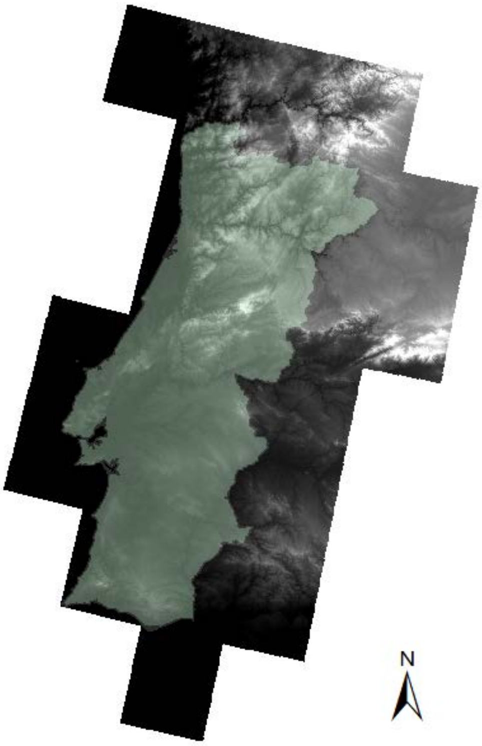


Figure 11: Elevation raster from ASTER GDEM for the study area.

The following R code was used to extract the elevation values from the raster image:

```r
#import Elevation raster
Elevation <- ”Elevation Raster Path”

#create raster for elevation
elevationLayer <- raster(Elevation)

# Extract the elevation raster values underlying the fire points
shp@data$elevationValues <- extract(elevationLayer, shp)
```

3.1.4 Meteorological Data : Temperature, Windspeed, Precipitation

The Agri4Cast system, also known as the Monitoring Agricultural Resources (MARS) Crop Yield Forecasting System is centered on the European commission’s Joint Research Centre (JRC). It provides the above mentioned meteorological data as CSV files. The data is obtained from many different weather stations across Europe. It is interpolated on a 25×25 km temporal grid (daily) ie. the data consists of a 25×25 km grid for every single day in the 16-year period. The mean air temperature, mean daily wind speed and the sum of precipitation values are available in Celsius, meters per second and millimeters per day respectively. The values from the the temporal gridded data were matched with the fire pixels using the nearest neighbour method and the following R code (example for precipitation):

```
#Reading the csv file
precipitationData <- read.table(”CSV file path”,sep = ”;”,header = TRUE)

#Conversion of date values. For example, converting integer 20000101 to date 2000-01-01
precipitationData$Date <- as.Date(as.character(precipitationData$DAY), format='%Y%m%d')

#AddÃŋng Precipitaion value for each point
for(j in 1:nrow(shp@data)){

#the values for the date of the fire pixel
 test <- subset(precipitationData, Date == shp@data[j,]$DATE)

#making it a spatial point data frame with 4326 projection(WGS_84)
 coordinates(test) <- c(”LONGITUDE”, ”LATITUDE”)
 proj4string(test)=CRS(”+init=epsg:4326”) # set it to lat-long

#transforming it to utm
  test = spTransform(test,proj4string(shp))

  #The fire pixel under observation
  thisPoint <- shp[j,]

 #which measurement is the nearest to our fire point ?
  nearestIndex <- apply(gDistance(test, thisPoint, byid=TRUE), 1, which.min)

  #finally, assigning the precipitation value, in mm
  shp@data$Precipitation[j] <- test[nearestIndex,]$Precipitation

}
```

Figure 12 illustrates the above mentioned method for a particular fire pixel.

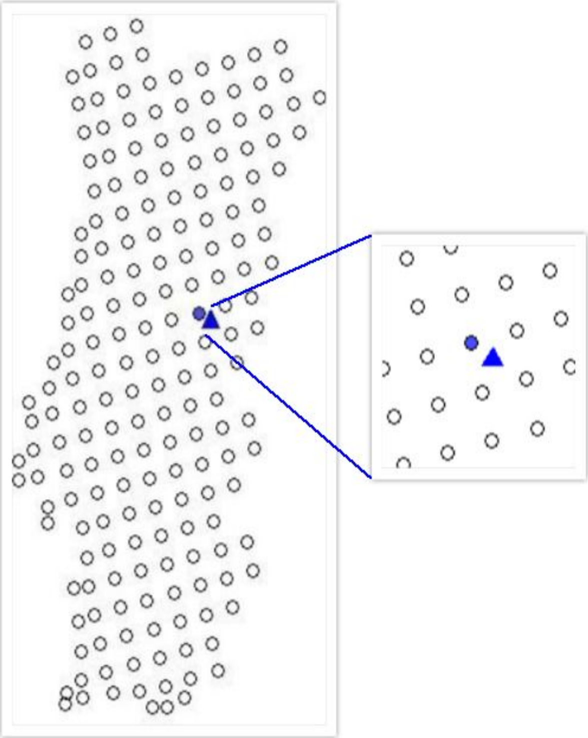


Figure 12: Extraction of meteorological data using nearest neighbour method.

The white circles are the weather grid values for Portugal on the day associated with the fire pixel. The blue triangle is the fire pixel's location. The nearest grid value to the fire pixel is depicted by the blue circle, which is assigned as the weather value for that fire pixel.

In the case of precipitation, it was observed that there was negligible precipitation associated with most of the fire pixels ie.there was no precipitation on most of the days when the fire pixels were detected.

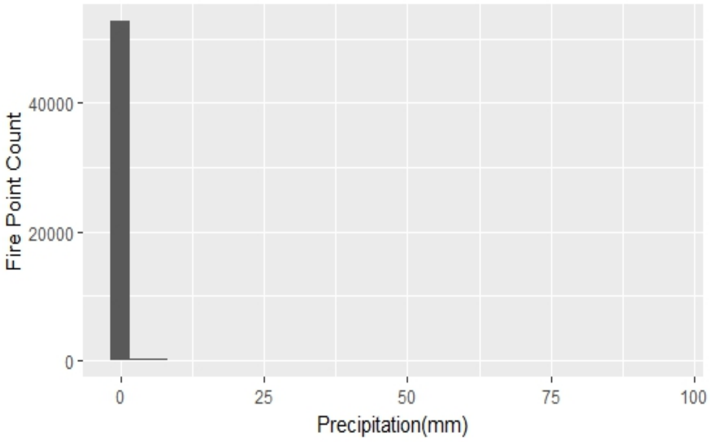


Figure 13: Precipitation (mm) associated with the fire pixels.

This is shown in Figure 13. This also proves that the precipitation is not a suitable indicator for forest fires in this study. Therefore, precipitation is rejected as an input parameter. It can be used to characterize a forest fire season as a time period with negligible precipitation, but that is not in the scope of this thesis.

3.1.5 Month of Fire Occurrence

As mentioned earlier, Portugal experiences a high number of forest fires in the summer season when compared to the winter. The frequency of the fire pixels was visualized over time. It was observed that the fires followed a seasonal pattern over the years, as shown in Figure 14. 2003 and 2005 are the years with most number of fire detections. In the box plot in Figure 15, it can be seen that July, August and September were the months where most of the fires were recorded, with some exceptions which will be discussed in the next chapter. Therefore, the month of fire occurrences was included as an input parameter for the one-class SVM model. The R code used to achieve the above mentioned time series analysis is given in Appendix B (A.2).

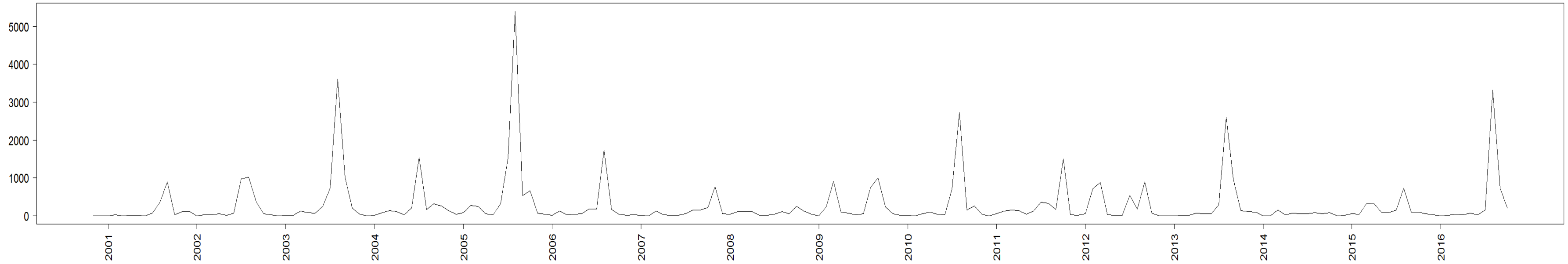


Figure 14: Frequency of fires over time.

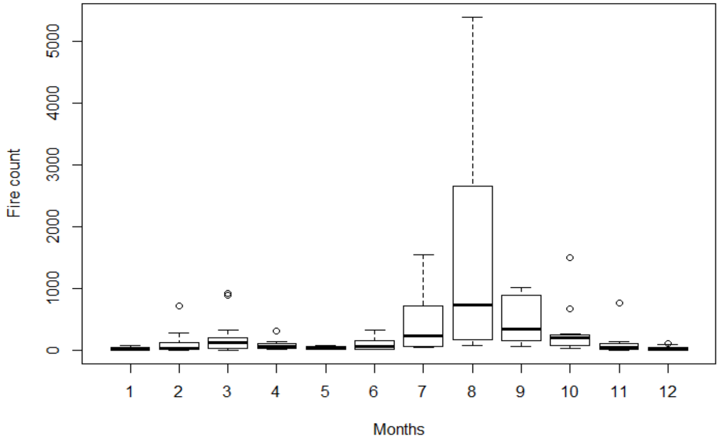


Figure 15: Monthly frequency of the fires.

3.1.6 Latitude

As visualized in Figure 9, it is clearly evident that most of the fires occurring in Portugal are in the northern and the central region. Figure 16 shows the kernel density map of the fires in Portugal, confirming that the density of fires is generally increasing with latitude with some exceptions which will be discussed in the next chapter. Therefore, latitude of the fire pixels was also included as an input parameter for the one-class SVM model.

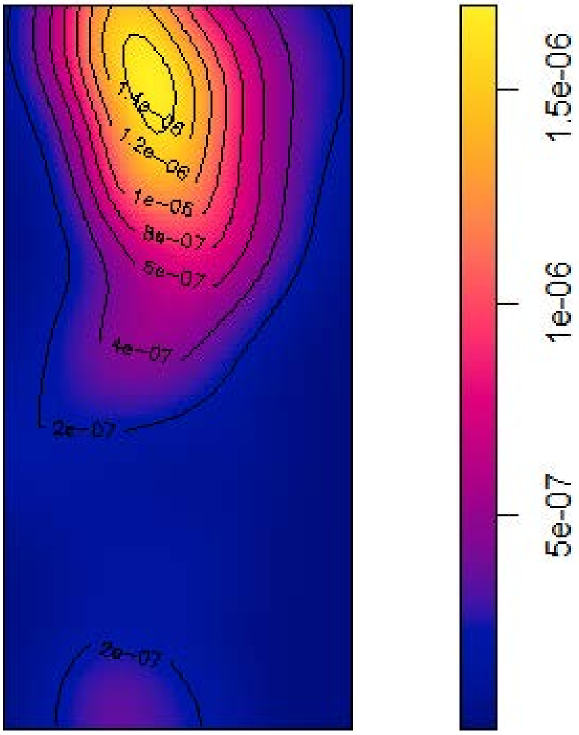
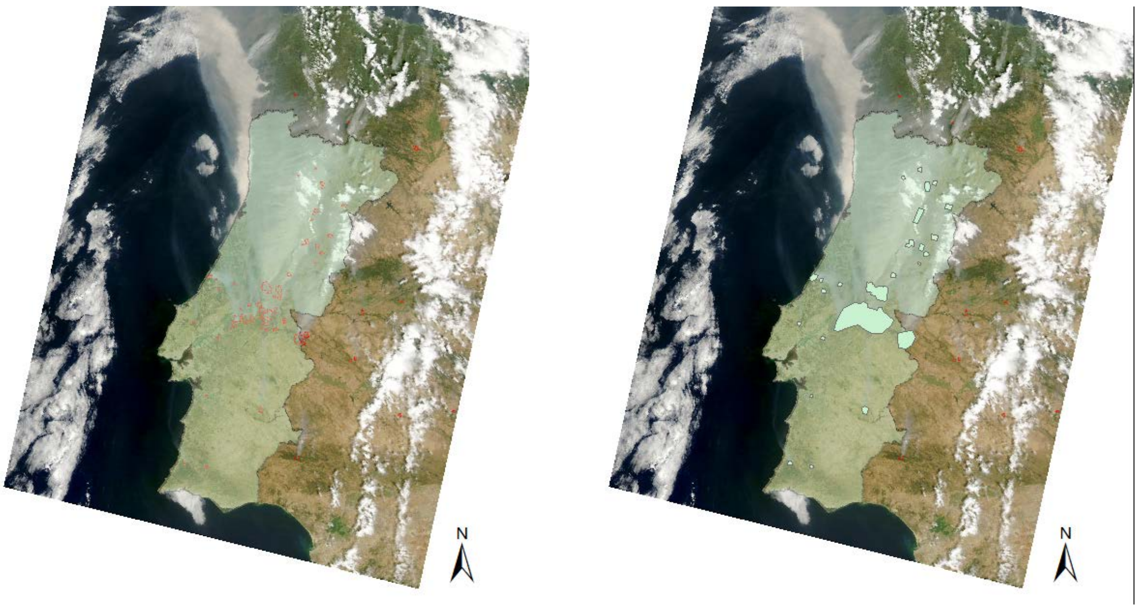


Figure 16: Kernel density map of the fire distribution in Portugal. R code can be found in appendix B (A.2).

3.2 Known Forest Fire Events: Collection and Outlier Removal

3.2.1 Collection

To train the one-class SVM model, fire pixels that belonged to forest fires had to be identified. For this, 38 known large forest fire events were identified through historical research. The images for these fire events were obtained from NASA's Earth Observatory and DLR's ZKI service. Figure 17a shows an example of a georeferenced forest fire image on QGIS.



(a) Georeferenced forest fire image. The fires are outlined in red.

(b) Digitizing the fire extents.

Figure 17: Extraction of known fire events.

The extent of these fires were digitzed as shown in Figure 17b. These digitized shapes were used to extract the respective fire pixels from the MODIS data. The fire pixels were matched with the shapes based on the date of the fire. The following R code was used for the extraction:

```r

## Add an Id to the shp data frame (primary key)
shp@data$ID <- 1:nrow(shp@data)

#Example of fires on August 3,2003
#converting the date column to a date object vector
shp@data$DATE <- as.Date( as.character(shp@data$ACQ_DATE))

#the vector used to store the IDs of the forest fire pixels
BigFiresIDs <- numeric()

#Function used to clip the fires using the digitized shapes
clipRealFires <- function(clipPath,layerName,fireDate) {

  clipShapeTemp <- readOGR(dsn = clipShape_path, layer = layerName)
#transforming the shape to UTM 29N projection
  clipShapeTemp <- spTransform(clipShapeTemp, proj4string(shp))

#clipping the fires spatially
  TempFires <- InitialShp[clipShapeTemp,]
  #subsetting the fires temporally, using the fire date
  TempFires <- TempFires[TempFires$DATE == fireDate,]
  return(TempFires$ID)
}

clipShape_path <- "digitzed shape path"

tempIDS<- clipRealFires(clipShape_path,"layer name","2003-08-03")

#Storing the IDs of the extracted fires
BigFiresIDs <- c(BigFiresIDs,tempIDS)

FiresTable <- shp[BigFiresIDs,]

#Storing the forest fires (labeled data) in a seperate data frame
FiresExportTable <- select(FiresTable@data,LATITUDE,LONGITUDE,DATE,ID,BRIGHTNESS,
       FRP, BRIGHT_T31,Temperature,fireMonth,WindSpeed,CONFIDENCE,
                               LULC_val,LanduseClass, elevationValues)
```

A total of 6767 fire pixels were identified as fire pixels belonging to large fires. Hereon, these fire pixels will be referred to as **“labeled samples/data”**. The rest of the fire pixels will be referred to as **“unlabeled samples/data”**. The pixels in the unlabeled data may or may not be large forest fire pixels. The aim is to locate the ones that are more likely to be large forest fire pixels in the unlabeled data. Figure 18 shows the spatial distribution of the labeled data. The blue circles resemble the fire pixels. Most of the fires are located in the north of Portugal.

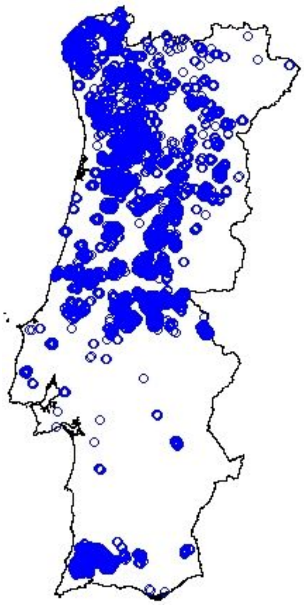


Figure 18: Spatial distribution of the labeled data.

Most of the fires occurred during summer in the month of August as shown below in Figure 19.

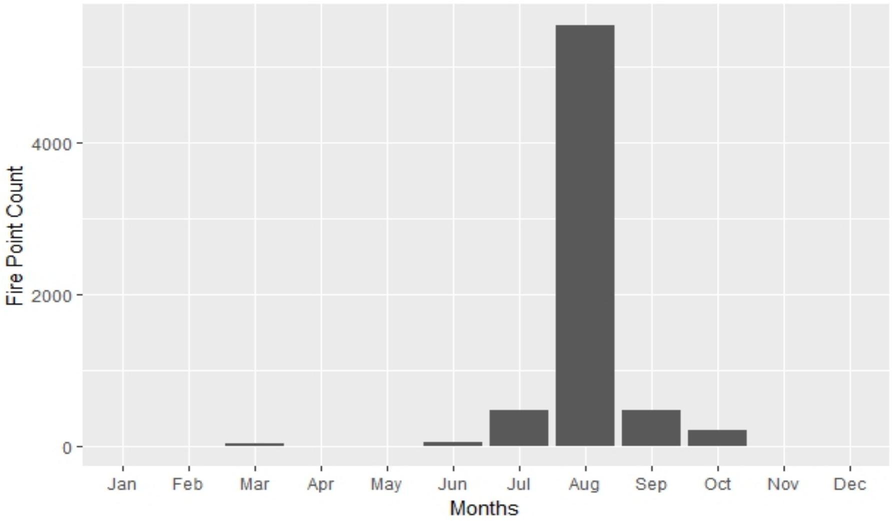
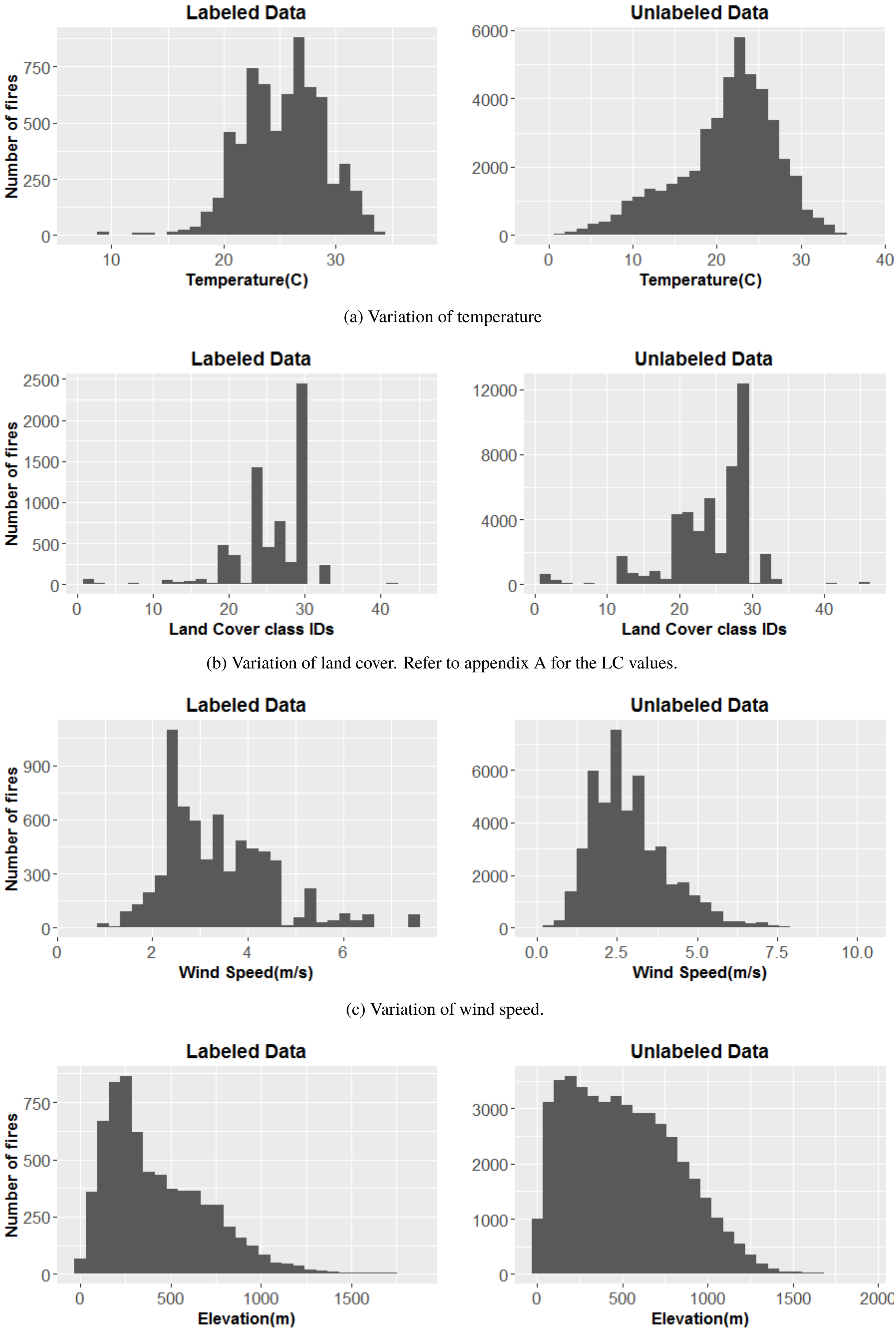


Figure 19: Monthly distribution of the labeled data.

3.2.2 Analysing and Visualizing the Input Parameters

The basic idea behind this thesis is that the fire pixels in the labeled data (large forest fires) exhibit certain unique characteristics. These characteristics could lead to the detection of similar fire pixels from the unlabeled data. Figure 20 compares different input parameters of the labeled data to that of the unlabeled data. Figure 21 shows similar graphs for the MODIS attributes. The distribution of input parameter values in the labeled data is different when compared to the unlabeled data. The values are more clustered in the labeled data when compared to the unlabeled data. In other words, the fire pixels in the labeled data follow certain patterns that set them apart from the unlabeled fire pixels. In the graphs for temperature, LC, windspeed, elevation and confidence, some outliers are visible that are caused probably by the digitization errors from the previous step ie. non-forest fire pixels were included in the labeled data due to human error while digitization (Figure 17b), measurement errors in the algorithm or noise. The graphs for BTs and FRP mostly have upper outliers, which are also discussed in the next sub-chapter.



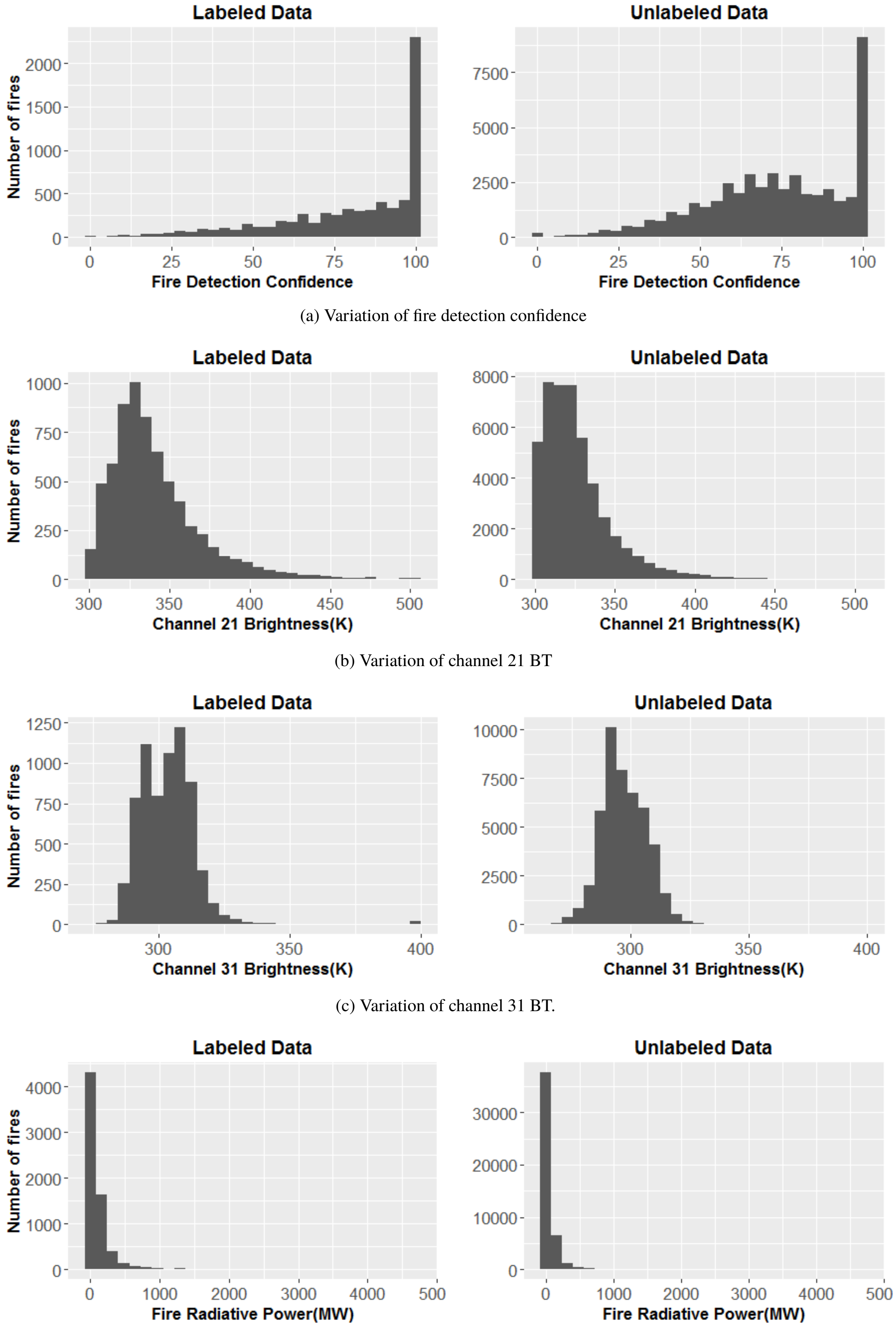
(a) Variation of temperature

(b) Variation of land cover. Refer to appendix A for the LC values.

(c) Variation of wind speed.

(d) Variation of elevation

Figure 20: Variation of input parameters.



(a) Variation of fire detection confidence

(b) Variation of channel 21 BT

(c) Variation of channel 31 BT.

(d) Variation of FRP

Figure 21: Variation of input parameters (MODIS attributes).

Tables 2 and 3 compare the labeled and unlabeled samples statistically in terms of mean and median. The values for both datasets are mostly different, which again shows that the large forest fire pixels do not exhibit the same characteristics as a general population of fire pixels. However, there is an exception in the case of BTs which will be discussed in chapter 6. In the case of land cover, it is a categorical variable. So, the graph in Figure 20 is a better method of visualization for land cover than the statistics given below.

	Input Parameter	Mean	Median
1	Temperature	25.33	25.5
2	Land Cover	25.49	27
3	Wind Speed	3.37	3.2
4	Elevation	427.98	347
5	Confidence	80.55	88
6	Channel 21 BT	340.88	334.1
7	Channel 31 BT	303.24	303.5
8	FRP	130.30	53.1

Table 2: Labeled data.

	Input Parameter	Mean	Median
1	Temperature	21.15	22.1
2	Land Cover	24.28	26
3	Wind Speed	2.88	2.7
4	Elevation	508.73	477
5	Confidence	73.09	75
6	Channel 21 BT	326.09	321.1
7	Channel 31 BT	297.25	296.2
8	FRP	66.17	27.5

Table 3: Unlabeled data.

3.2.3 Outlier Analysis

A box plot outlier analysis is performed to detect the outliers found in the previous step. The box plots give the minimum and the maximum thresholds for removing the outliers as explained in section 2.2.2. The labeled data basically represents a class (large forest fires). By removing the extreme outliers, the definition of this class is made more specific by cutting down the range of values ie. the class is now defined by points that are less scattered than before in the feature space (refer 2.4.1). The R code used to obtain the thresholds is given in Appendix B (A.2). The 5 summary values: Minimum, First Quartile (Q_1), Median(Q_2), Third Quartile(Q_3) and Maximum, are mentioned in Table 4. The minimum and the maximum values are the thresholds used.

	Input Parameter	Minimum	Q_1	Median	Q_3	Maximum
1	Temperature	15.5	22.8	25.5	28.1	33.6
2	Elevation	7	207	347	614	1223
3	Wind Speed	0.5	2.5	3.2	4.1	6.4
4	Land Cover	14	23	27	29	33
5	Confidence	20	68	88	100	100
6	FRP	3	24.6	53.1	121.25	266.2
7	Channel 21 BT	300	321.9	334.1	352.5	398.4
8	Channel 31 BT	276	295.4	303.5	309.5	330.5

Table 4: Box plot summary values.

The following code was used to apply the thresholds to the labeled data :

```r
#Applying the thresholds
fires_withoutOutliers <- subset(FiresExportTable, BRIGHT_T31 > 275.99)
fires_withoutOutliers <- subset(fires_withoutOutliers, FRP > 2.99)
fires_withoutOutliers <- subset(fires_withoutOutliers, BRIGHTNESS > 299.99)
fires_withoutOutliers <- subset(fires_withoutOutliers, Temperature > 15.4 & Temperature < 33.7)
fires_withoutOutliers <- subset(fires_withoutOutliers, WindSpeed > 0.4 & WindSpeed < 6.5)
fires_withoutOutliers <- subset(fires_withoutOutliers, CONFIDENCE > 19)
fires_withoutOutliers <- subset(fires_withoutOutliers, LULC_val > 13 & LULC_val < 34)
fires_withoutOutliers <- subset(fires_withoutOutliers, elevationValues > 6 & elevationValues < 1223 )

#Re-exporting the tables without outliers
BigFiresIDs <- fires_withoutOutliers$ID

#number of big fires after removing the outliers --> 6206
length(BigFiresIDs)

FiresTable <- shp[BigFiresIDs ,]

#Forest Fires
FiresExportTable <- select(FiresTable@data,LATITUDE,LONGITUDE,DATE,ID,BRIGHTNESS,
        FRP, BRIGHT_T31,Temperature,fireMonth,WindSpeed,CONFIDENCE,
                                   LULC_val,LanduseClass, elevationValues)

NonFireTable <- setdiff(shp@data,FiresTable@data )

#rest of the data
NonFiresExportTable <- select(NonFireTable,LATITUDE,LONGITUDE,DATE,ID,BRIGHTNESS,
          FRP,  BRIGHT_T31,Temperature,fireMonth,WindSpeed,CONFIDENCE,
                                     LULC_val,LanduseClass, elevationValues)
```

Around 450 fire pixels were removed from the labeled data through this analysis. Figure 22 shows the visualization of the outliers for different input parameters. For FRP and BTs, only the lower outliers are removed because high values of BTs are essential criteria for the C6 algorithm for active fire detection and high FRP is an ovbious indication of a fire. These values cannot be considered as outliers. Also, latitude and month of fire occurrence are not a part of this analysis. Because latitude would result in removal of the known fire events in the south of Portugal and the month parameter would result in removal of fires from non-summer months as most of the known fires occurred in July and August and in Northern and Central Portugal (explained in detail in the next chapter).

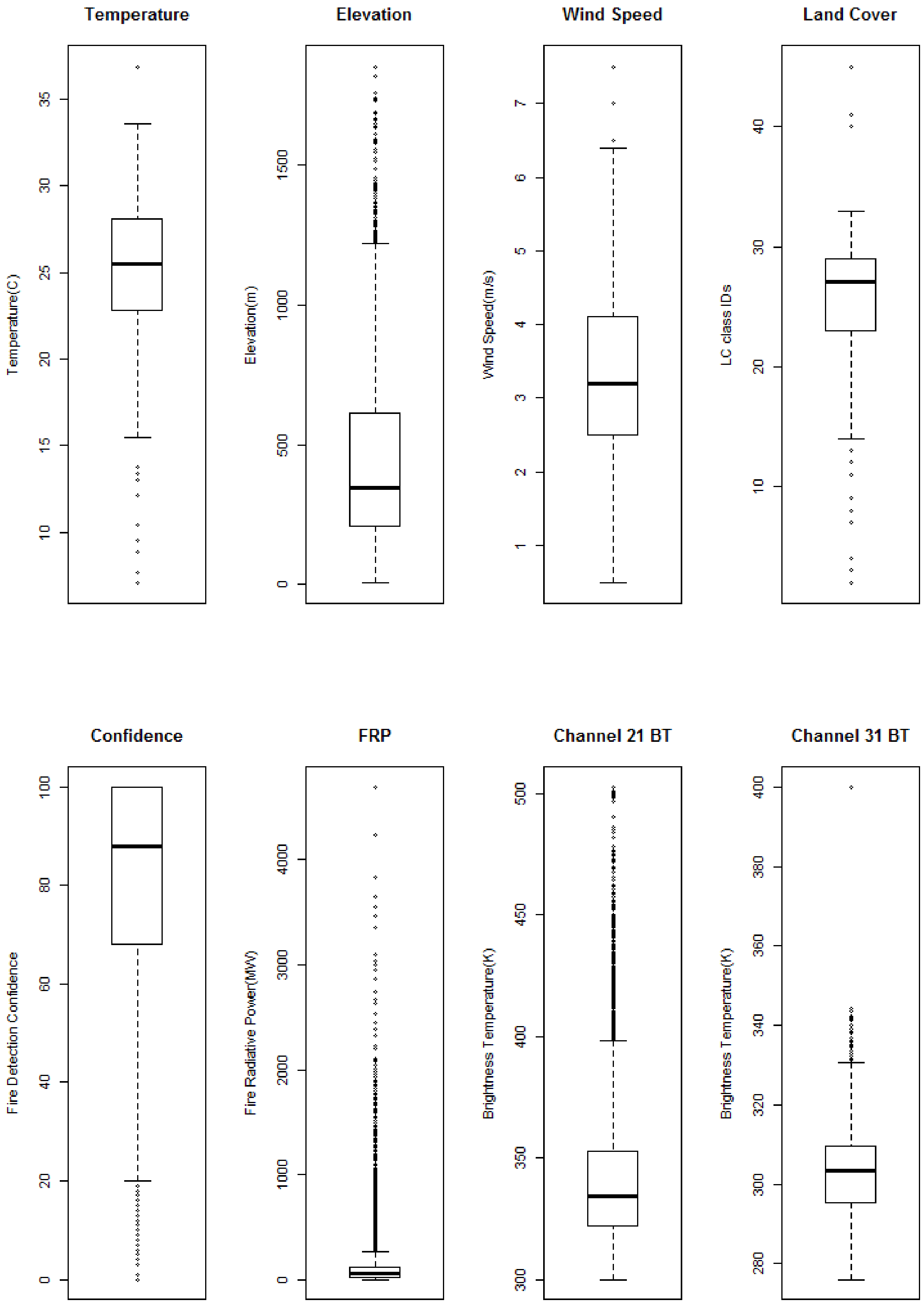


Figure 22: Visualization of outliers for different input parameters.

3.3 Known Forest Fire Events: Cluster Analyses

3.3.1 Clustering Behavior of Forest Fires

The nearest neighbour method explained in section 2.2.1 is used to estimate the clustering degree of the known forest fire events (labeled data). The fire events were considered as separate distributions and NNI was calculated for each one of them. Area for all the indices was calculated using a bounding box. Also, the Average Nearest Neighbour Distance (ANND) for each distribution was obtained using the following code:

```r
#Get Big Fire IDs from outlier analysis
RealFirePoints <- shp[BigFiresIDs ,]

#data frames to store the NNI and average nearest neighbour distance
avgNND_perDay<- data.frame(avg_NND = numeric())

NNI_perDay <- data.frame(NNI = numeric())

for(realDate in 1:length(unique(RealFirePoints@data$DATE))){
  #Fires from the current fire event
  thisDayFires <- RealFirePoints[RealFirePoints$DATE ==
                                   unique(RealFirePoints@data$DATE)[realDate],]

  #Do it only if there are atleast 2 points in the same overpass (Area calculation)
  if(nrow(thisDayFires) > 2){

    #x and y coordinates
    x <- coordinates(thisDayFires)[,1]
    y <- coordinates(thisDayFires)[,2]

    #coordinates for the bounding box
    l1 <- bbox(thisDayFires)[1,1]#x
    l2 <- bbox(thisDayFires)[1,2]#x
    l3 <- bbox(thisDayFires)[2,1]#y
    l4 <- bbox(thisDayFires)[2,2]#y

    #window of observation
    win <- owin(xrange=c(l1,l2), yrange=c(l3,l4))

    #converting to ppp
    day_ppp <- ppp(x, y, window=win, marks=thisDayFires@data)

    ##Nearest neighbour index test
    #n = number of points
    n = length(nndist(day_ppp))

    #D = mean nearest neighbour distance
    D = mean(nndist(day_ppp))
    avgNND_perDay[realDate,] <- c(as.numeric(D))

    #Bounding Box Area
    A=area.owin(bounding.box.xy(coords(day_ppp)))

    #NNI
    R=2*D*sqrt(n/A)
    NNI_perDay[realDate,] <- c(as.numeric(R))

  }
}

#NNI
mean(NNI_perDay$NNI, na.rm= TRUE)
#[1] 0.27

#cluster size
mean(avgNND_perDay$avg_NND, na.rm= TRUE)
#[1] 1328.82
max(avgNND_perDay$avg_NND, na.rm= TRUE)
#[1] 3057.753
```

The average NNI was 0.27 which means that most of the forest fire events were clustered. The mean ANND is ∼1328 meters and the maximum is ∼3058 meters. This means that the most dispersed fire event has an ANND of 3.058 km. This radius would ensure that all the fire events are considered while computing the cluster parameters in the next step which depend on the number of spatial and temporal neighbours. Therefore, 3058 meters was used as the radius for spatial neighbour search in the next step.

3.3.2 Spatial and Temporal Cluster Analysis

The objective of this analysis was to quantify the spatial and temporal cluster behaviour of forest fires. Figure 23 illustrates the concept behind the calculation of the spatial and ST parameters for the fire pixels. A spatial filter of x meters is applied to reject pixels (highlighted in red) that

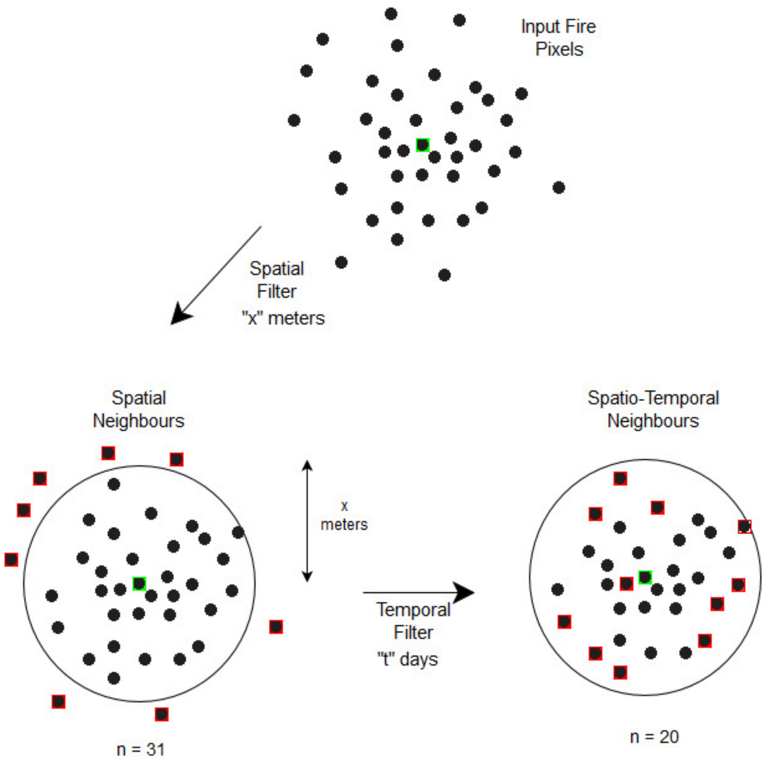
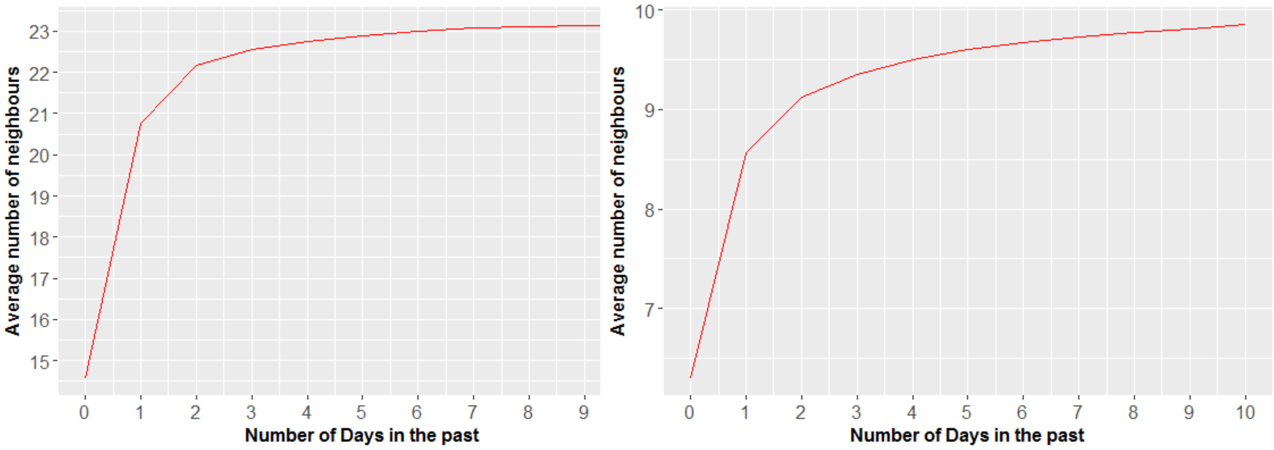


Figure 23: Concept behind the cluster parameters.

are out of spatial proximity of the observed pixel (highlighted in green). The value of x here is 3058 meters, obtained from the previous step. This is the radius that is used to count the number of spatial neighbours for each fire pixel. This is the first cluster parameter, which is equal to 31 in this case.

Then, a temporal filter of n days is applied to reject the fire pixels that are temporally not close enough to the observed fire pixel. This results in another cluster parameter which is equal to 21 in this case, as shown in the figure above. This parameter has both spatial and temporal components.



(a) Labeled Data

(b) All Data.

Figure 24: Variation of ST neighbours with increasing number of days in the past.

To quantify the temporal characteristics of the fires, it is essential to get the right estimate for the value of n. For this, each pixel in the labeled data was examined in terms of ST neighbours with increasing number of days in the past (Figure 24a). The average number of neighbours in a radius of 3058 meters were plotted against the number of days (in the past) under observation. The graphs show the cumulative sum on the y-axis ie. for n = 2 (in Figure 24a), the value on y axis is 22 which means that the y-value is 22 for 2 days together and not the 2nd day alone. It can also be seen that the y-value for the labeled data is generally much greater than that for the data including all the fire pixels (Figure 24b), which means that the forest fire pixels are more clustered when compared to the whole dataset. The y-value in the in fig 24a experiences a sharp jump from n=0 to n=1 and then, this is more or less constant. It is 14 for the day of the fire occurrence (n=0), ∼21 for one day in the past and then it is 22∼23. This means that the day of the fire occurrence and the previous day are the most critical days for temporal clustering signified by the jump. Therefore, two new parameters are introduced: number ST neighbours on the day of the fire occurrence and in a 2-day window. The code that is used to perform this analysis is given in Appendix B (A.2). On the whole, one spatial parameter and two ST parameters have been introduced in this sub-chapter using the following code:

```
#create x meters buffers around each point
shp_buffer <- gBuffer(shp, width = 3058, byid = TRUE)

## For each ID, find IDs within that Buffer
#temporary dataframe bak
bak <- shp

#Using only the relevant columns
bak@data <- select(bak@data, ID, DATE)

#overlaying these buffers on the points. Each buffer is linked to a fire point through the ID
bak_over <- over(shp_buffer, bak, returnList = TRUE)

#data frame with count of spatial and spatio temporal neighbors
res <- data.frame(circleID = numeric(), ct_SpTempNeighbors = numeric(),ct_SpatialNeighbors =
    numeric()
                  ,ct_SpatialNeighbors_sameDay = numeric())

#Processing each buffer
for(j in 1:length(bak_over)){

  tmp <- bak_over[j]

  #list to dataframe
  df <- ldply (tmp, data.frame) # http://stackoverflow.com/a/4227483

  #The ID of the buffer origin
  CircleName <- names(tmp)
  df$CircleID <- CircleName

  #Date of the buffer origin
  CircleDate <- df$DATE[which(df$ID == CircleName)] # or df$CircleID instead of circle name

  #Removing the origin, because we don't want it to be considered as a neighbour of itself
  df <- filter(df, ID != CircleName) #filtering out the ID itself, the fire point

  #timedifference, between the date of fire and the date of neighbors, in days
  df$TIMEDIFF <-difftime(df$DATE, CircleDate, units = ”days”)

  #filtering out the points that are not spatially close enough (n = 1)
  df_timeFilter <- filter(df, TIMEDIFF >= -1 & TIMEDIFF <=0)

  #filtering out the points that are not on the same day  (n = 0)
  df_timeFilter_sameDay <- filter(df, TIMEDIFF == 0)

  #storing the count, number of spatio temporal neighbors, spatial neighbors, spatial neghbors
     on same day
  res[j,] <- c(as.numeric(CircleName), nrow(df_timeFilter), nrow(df), nrow(df_timeFilter_
    sameDay))
}

#Joining on the basis of ID
shp@data <- dplyr::left_join(shp@data,res,by = c(”ID”=”circleID”))
```

In this thesis, it is assumed that a pixel needs to have at least one spatial or ST neighbour to qualify as a true forest fire pixel. Therefore, an additional filter is applied to remove the pixels from the labeled data that are spatially and spatio-temporally separated. This is done in order to introduce a clustering condition to the one-class SVM model through the labeled data. The code to apply the cluster filters is given below:

```r
#ST neighbours for n=0,1
fires_withoutOutliers <- subset(FiresExportTable,ct_SpTempNeighbors > 0 )
fires_withoutOutliers <- subset(fires_withoutOutliers , ct_SpatialNeighbors_sameDay > 0 )
#Spatial neighbours
fires_withoutOutliers <- subset(fires_withoutOutliers , ct_SpatialNeighbors > 0 )
#new Big Fires
BigFiresIDs <- fires_withoutOutliers$ID

#number of big fires after removing the outliers --> 6102
length(BigFiresIDs)

#re exporting the labeled data
FiresTable <- shp[BigFiresIDs,]

#Big Fires
FiresExportTable <- select(FiresTable@data,LATITUDE,LONGITUDE,DATE,ID,BRIGHTNESS,FRP,
                           BRIGHT_T31,Temperature,fireMonth,WindSpeed,CONFIDENCE,
                           LULC_val, elevationValues,ct_SpTempNeighbors,
                           ct_SpatialNeighbors_sameDay, ct_SpatialNeighbors)


NonFireTable <- setdiff(shp@data,FiresTable@data )

#rest of the data
NonFiresExportTable <- select(NonFireTable,LATITUDE,LONGITUDE,DATE,ID,BRIGHTNESS,FRP,
                              BRIGHT_T31,Temperature,fireMonth,WindSpeed,CONFIDENCE,
                              LULC_val, elevationValues,ct_SpTempNeighbors,
                              ct_SpatialNeighbors_sameDay, ct_SpatialNeighbors)
```

All in all, 665 outliers were removed from the labeled data. Figure 25 shows the labeled data before and after the outlier analysis, highlighting the outliers in red.

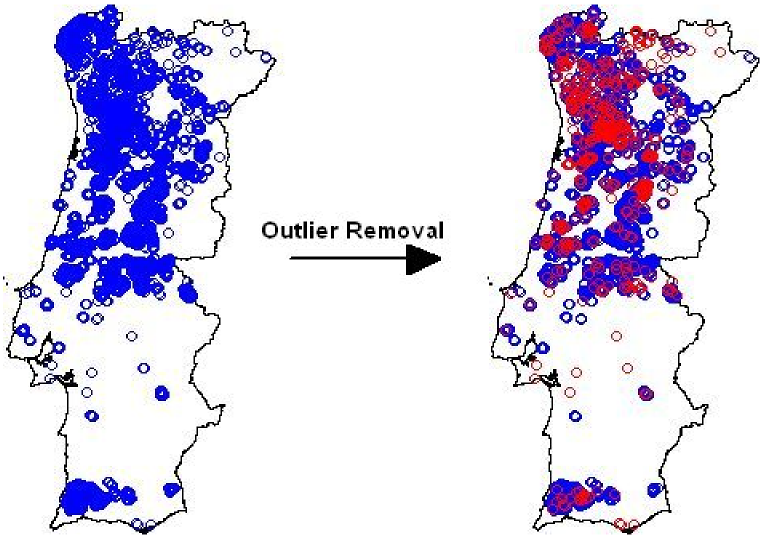


Figure 25: Labeled data: before and after outlier analysis.

3.4 Summary

In this chapter, a total of 13 input parameters have been described for the one-class SVM model. The overview of the input table is given in Table 5.

Fire Pixels	Latitude	Month	Channel 21 BT	FRP	Channel 31 BT	Temperature	Wind Speed	Confidence	Land Cover	Elevation	Spatial Neighbours	ST Neighbours n = 0	ST Neighbours n = 1
1	41.11	8	335.20	88.00	307.90	30.80	2.20	84	21.00	400.00	9.00	9.00	16.00
2	41.10	8	332.00	71.40	307.60	30.80	2.20	80	28.00	372.00	11.00	11.00	21.00
3	40.51	8	331.50	88.40	300.70	29.10	2.20	80	29.00	901.00	13.00	13.00	43.00
4	40.51	8	354.60	287.50	303.20	29.20	2.60	97	27.00	829.00	6.00	6.00	31.00
5	40.50	8	352.00	243.40	302.10	29.10	2.20	96	21.00	897.00	12.00	10.00	45.00
.													
.													
.													
.													
.													
.													
n													

Table 5: Input table

The R code used to create the above table is given in Appendix C (A.3). Out of 53331 fire pixels, 6102 are a part of the labeled dataset and the rest are unlabeled. In the next chapter, the labeled samples are passed as input to a one-class SVM and the resulting model is used to detect similar fire pixels that are likely to be a part of large fires from the unlabeled dataset.

4 APPLICATION OF ONE-CLASS SVMs

This chapter explains the application of the input parameters (derived in the previous chapter) by a one-class SVM model. The first sub-chapter gives the general workflow for the application of a one-class SVM in this thesis. The second sub-chapter explains the tuning process involved in order to obtain the best results from the one-class SVM model. The third sub-chapter describes the techniques used to prevent overfitting of the model.

4.1 Basic Workflow

In the previous chapter, the MODIS hotspots were split into 2 parts: labeled and unlabeled data. As explained earlier in sub-chapter 2.4.1, the one-class SVM requires a known or labeled dataset that is a collection of vectors belonging to a single class. The labeled data, now consists of 13 different input parameters / attributes that define the fire pixels that belong to large forest fires. Figure 26 illustrates the basic workflow for the application of a one-class SVM in this thesis.

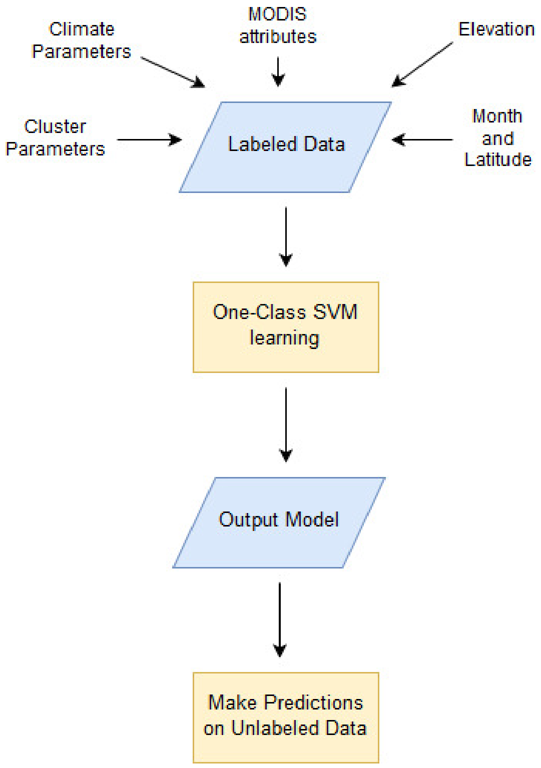


Figure 26: Application of a one-class SVM.

The one-class SVM produces a model as an output after learning the classes in the labeled data. The learning process needs to be performed with the optimum hyperparameters for the model, which is described in the next sub-chapter. The model is then used to make logical predictions (TRUE/FALSE) on the fire pixels in the unlabeled data. Based on the prediction results, the model can also be corrected for overfitting, which is done and described in 4.3. The R library *e1071* provides the required functions to create and tune the one-class SVM model for this thesis.

4.2 Tuning the one-class SVM model

In chapter 2.4.1, it was mentioned that the SVMs are highly dependent on some so called *hyperparameters*. These hyperparameters are Cost, Gamma (kernel parameter) and Nu. A slight modification in any of these parameters might result in huge errors in the predictions. The *e1071* package provides a tune function that can be used to tune the SVM model to obtain the optimum hyperparameters. It performs a grid search and a 10-fold CV to select the best parameters for the model. For one-class SVMs, only nu and gamma need to be estimated (as explained in 2.4.1). An example of this function is given below :

```
#Tuning the model, x contains the input parameters and y contains a vector with class value.
#In this case, y is  logical Vector with values = TRUE (single class)
tuned <- tune.svm(x=x,y=y,
                  nu =  0.001:1.0,
                  gamma = 10^(-4:0),
                  type='one-classification')
```

The lower and upper grid limits of each parameter are passed to the tune function along with a type “one-classification” to specify that it is done for a one-class SVM model. In this case, a 2-dimensional grid is created because of the number of parameters. A model is created for each combiination (γ,nu) in this grid. 10-fold CV is used as the sampling method for each model. Figure 27 illustrates a 10-fold CV process. The labeled data is randomly split into 10 equal parts. 9 parts are used as training set and the 10th part is used for testing. The CV step is repeated 9 more times, with each part used as the test data exactly once. The 10 results are averaged to produce an estimation. These results are classification errors ie. the fraction of the test data that was predicted incorrectly.

The tune function looks for the model with the lowest classification error. The parameters of that model are given as output as shown in the example below:

```
tuned

#Parameter tuning of svm:
#- sampling method: 10-fold cross validation
#- best parameters:
# gamma    nu
# 0.001    0.001
#- best performance: 0.002295082. This is the error value of the best model
```

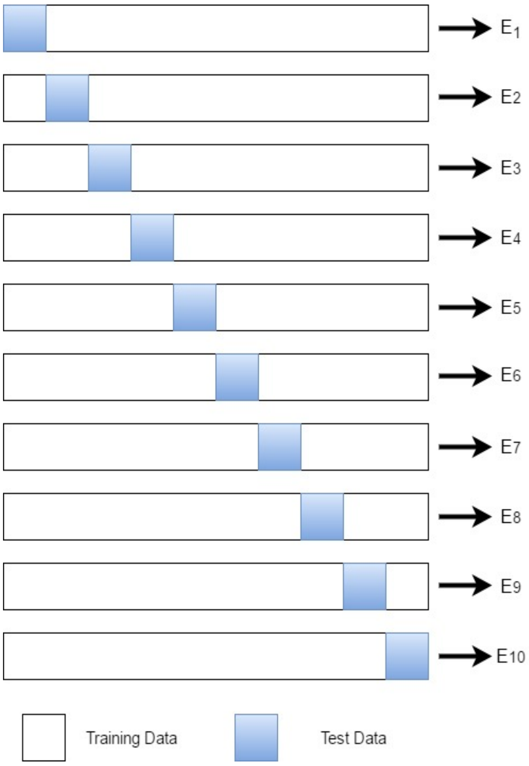


Figure 27: 10-fold cross-validation.

4.3 Cross-Validation

Overfitting is often caused due to high model complexity. This happens when there are too many input parameters defining the one-class distribution (labeled data). This makes the model too specific to the input/labeled data. The aim is to detect similar fire pixels from the unlabeled data and not identical ones. The following techniques are used in this thesis to test the model for overfitting.

4.3.1 Leave-one-out Cross-Validation

The objective of this method is to analyse the influence of a single large fire event on the one-class SVM model. In this CV technique, each large fire event is left out of the labeled data and a model is created using the other fire events as training set. This model is used to detect the left out fire event and the accuracy of the predictions is examined. If $(F_1,F_2,F_3,F_4.....F_n)$ are the large fire events, F_1 is left out in the first run and $(F_2,F_3,F_4.....F_n)$ are considered as the training set to build the model and the predictions are made on F_1. This step is repeated n times for all the events. The difference between this approach and the 10-fold CV approach is in this case, that the labeled data is split in terms of the fire events opposed to a random split in 10-fold CV.

The prediction results are given in Table 6.

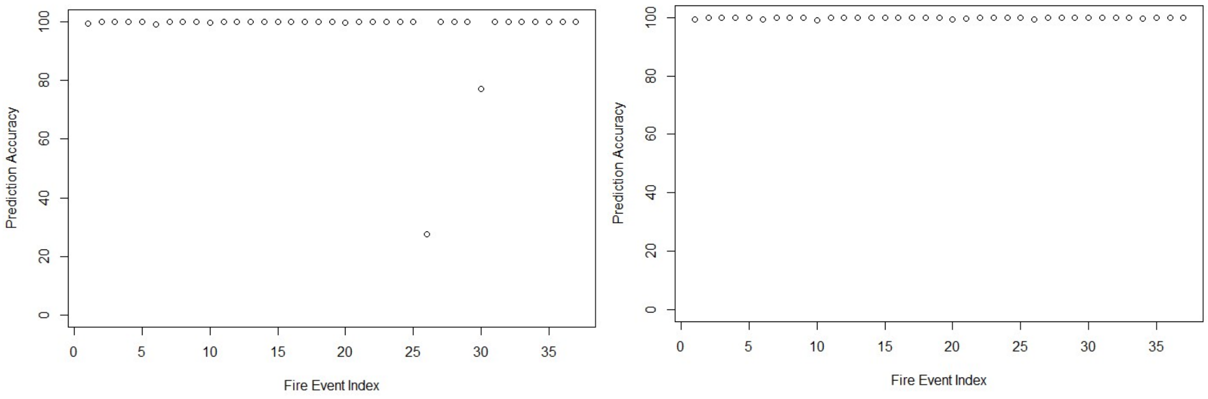
Index	Prediction Accuracy	Fire Date
1	99.41	2003-08-03
2	100	2003-08-04
3	100	2003-08-07
4	100	2003-08-08
5	100	2003-08-09
6	99.21	2003-08-11
7	100	2003-08-12
8	100	2003-08-13
9	100	2003-07-28
10	99.6	2003-09-12
11	100	2003-09-13
12	100	2004-07-26
13	100	2004-07-28
14	100	2005-08-04
15	100	2005-08-05
16	100	2005-08-06
17	100	2005-08-07
18	100	2005-08-14
19	100	2005-08-16
20	99.78	2005-08-21
21	100	2005-08-22
22	100	2005-08-24
23	100	2005-07-12
24	100	2005-07-20
25	100	2005-07-23
26	27.6	2005-10-03
27	100	2006-08-13
28	100	2006-08-07
29	100	2006-08-09
30	77.14	2006-06-05
31	100	2010-08-13
32	100	2013-08-29
33	100	2016-08-10
34	100	2016-08-11
35	100	2016-08-12
36	100	2016-08-13
37	100	2016-08-14

Table 6: Prediction accuracy.

Index	Prediction Accuracy	Fire Date
1	99.26	2003-08-03
2	100	2003-08-04
3	100	2003-08-07
4	100	2003-08-08
5	100	2003-08-09
6	99.21	2003-08-11
7	100	2003-08-12
8	100	2003-08-13
9	100	2003-07-28
10	99.2	2003-09-12
11	100	2003-09-13
12	100	2004-07-26
13	100	2004-07-28
14	100	2005-08-04
15	100	2005-08-05
16	100	2005-08-06
17	100	2005-08-07
18	100	2005-08-14
19	100	2005-08-16
20	99.34	2005-08-21
21	99.57	2005-08-22
22	100	2005-08-24
23	100	2005-07-12
24	100	2005-07-20
25	100	2005-07-23
26	99.39	2005-10-03
27	100	2006-08-13
28	100	2006-08-07
29	100	2006-08-09
30	100	2006-06-05
31	100	2010-08-13
32	100	2013-08-29
33	100	2016-08-10
34	99.70	2016-08-11
35	100	2016-08-12
36	100	2016-08-13
37	100	2016-08-14

Table 7: Prediction accuracy without month of fire occurrence.

These tables are represented graphically in Figure 28. Prediction accuracy is the fraction of fire pixels that were predicted correctly for each fire event that was left out. In Figure 28a, it can be seen that the prediction accuracy is almost 100 % for all the cases except for all the events except for event indices 26 and 30 where it is ∼20 % and ∼80 % respectively. These fire events occurred in the months of October and June (refer Table 6). Also, these two events are the only ones in the known labeled data that occur in these months. This means that the model rejects fire events from the months that are not in the usual fire season (July, August, September). This is caused by the input parameter “Month of Fire Occurrence”, which makes the model overfit with respect to the months in the labeled data. Therefore, this input parameter is removed and the whole process is repeated. The results are shown in Table 7 and graphically represented in Figure 28b.



(a) Prediction accuracy.

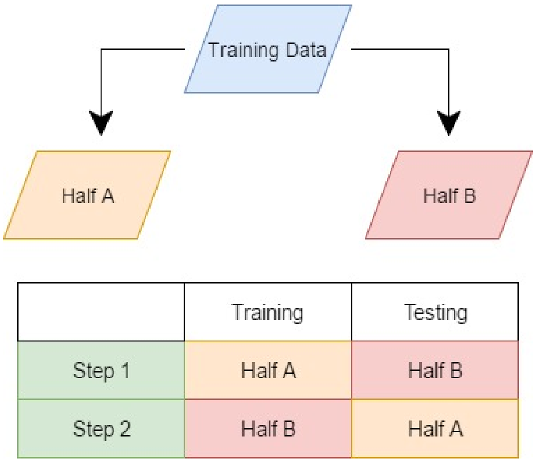
(b) Prediction accuracy without month of fire occurrence.

Figure 28: Prediction Accuracies.

As shown, the prediction accuracy is now almost 100 % for all the events after removing the month parameter. Therefore, the month parameter is rejected for the one-class SVM model. This method basically proves that an individual fire event does not affect the model in terms of prediction accuracy ie. the left-out event can be predicted correctly with the help of other fire events. The R code for the above analysis is given in Appendix D (A.4).

4.3.2 Hold-out Cross-Validation

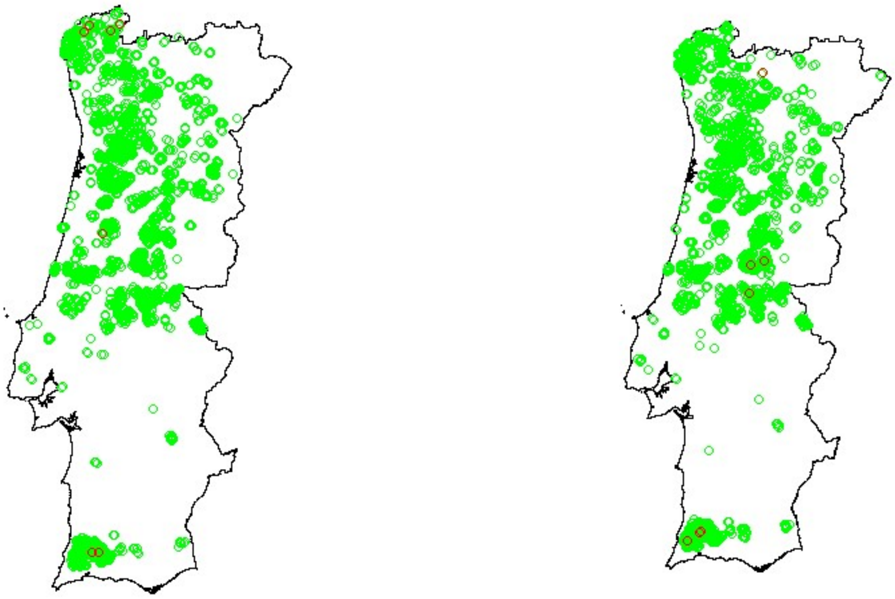
Figure 29 illustrates the Hold-out CV method concisely. The labeled data is split into 2 random halves: half A and half B. In the first step, half A is used to train the model and half B is used for testing. In the second step, half B is used to train the model and half A is used for testing. The prediction accuracies in both cases are examined. For an ideal case, the accuracies in both steps should be high. This method checks the dependence of the model on each half of the data.



	Training	Testing
Step 1	Half A	Half B
Step 2	Half B	Half A

Figure 29: Hold-out cross-validation.

The R code used to achieve the above mentioned steps is given in Appendix D (A.4). The prediction accuracies for both steps were almost 100 %. Figure 30 gives a pictorial description of the results. The green circles depict the true detections and the red circles depict the false detections.



(a) Step 1.

(b) Step 2.

Figure 30: Hold-out cross-validation results

It can be seen that the labeled set does not consist of fires in all regions of Portugal. The latitude parameter is the only input parameter that is dependent on the spatial locaiton of the fire pixels. This parameter might cause the model to reject fires in regions where there are no labeled samples (discussed in the next chapter).

Both the CV techniques used in this thesis prevent overfitting. Underfitting occurs when the model becomes too generic. For example, a simple/general classification condition might make the model predict too many fire pixels as true detections. In this thesis, validating a model for underfitting is not possible with the available data. Because the data does not consist of fire pixels labeled as false detections. This was the primary reason behind choosing a one-class SVM model, because only the data for true detections (large fire events) was available.

Finally, the one-class SVM model was derived using the following input table (with 12 parameters):

Fire Pixels	Latitude	Channel 21 BT	FRP	Channel 31 BT	Temperature	Wind Speed	Confidence	Land Cover	Elevation	Spatial Neighbours	ST Neighbours n = 0	ST Neighbours n = 1
1	41.11	335.20	88.00	307.90	30.80	2.20	84	21.00	400.00	9.00	9.00	16.00
2	41.10	332.00	71.40	307.60	30.80	2.20	80	28.00	372.00	11.00	11.00	21.00
3	40.51	331.50	88.40	300.70	29.10	2.20	80	29.00	901.00	13.00	13.00	43.00
4	40.51	354.60	287.50	303.20	29.20	2.60	97	27.00	829.00	6.00	6.00	31.00
5	40.50	352.00	243.40	302.10	29.10	2.20	96	21.00	897.00	12.00	10.00	45.00
.												
.												
.												
.												
.												
.												
n												

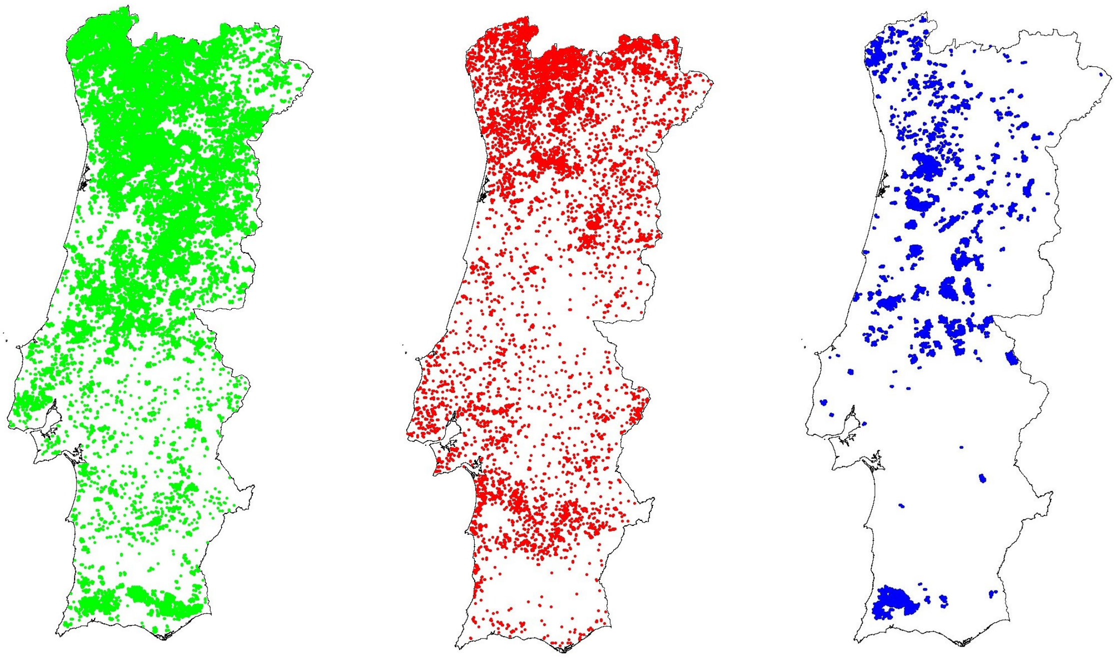
Table 8: Final Input Table

The model is applied on the unlabeled samples to make predictions. The results are described in the next chapter.

5 RESULTS AND DISCUSSIONS

5.1 Detection of Large Fire Events Using the One-Class SVM Model

The final one-class SVM model derived in the previous chapter was applied on the unlabeled samples to detect large fire events in the time period 2000-2016. The R code for the application of the model is given in Appendix E (A.5). As a result, 79.8 % of the fires were detected as true large fires. Figure 31 shows the spatial distribution of the true and false detections.



(a) True detections.

(b) False detections.

(c) Labeled samples.

Figure 31: One-class SVM model results.

To check the model's spatial dependence on the labeled samples (mentioned in 4.3.2), the results are compared with the spatial distribution of the labeled samples in the above figure. There is no identifiable region where the model shows excessive spatial dependence on latitude. There are no regions where there are only true detections caused by the presence of labeled samples in that region or only false detections caused by the absence of labeled samples in that region. This shows at a first glance, that the model is not excessively dependent on the latitude of the fire pixels.

To verify the effect of the tuning function described in 4.2, the model was also created with the default hyperparameters in R. Almost 3000 samples were used as support vectors by default which led to a true detection percentage of $\sim$22 %, which is clearly a case of overfitting.

It is already stated that the primary objective of this thesis is to increase the significance of true large fire events and not to identify the false detections. The true detections were visualized over time as shown in Figure 32.

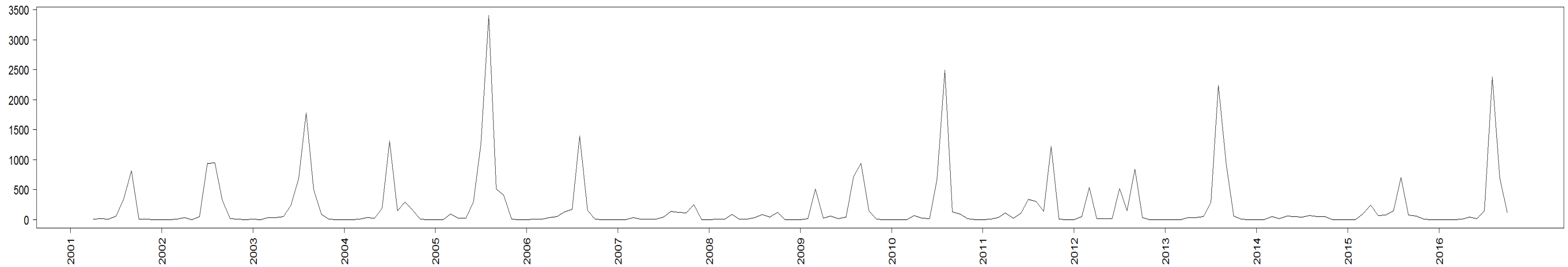
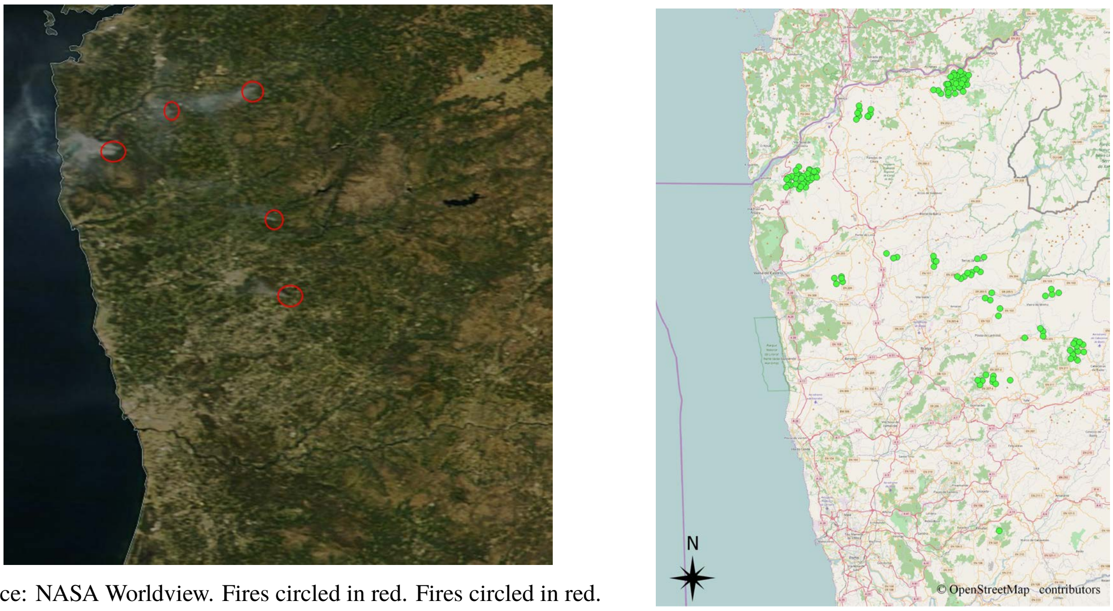


Figure 32: Frequency of true detections over time.

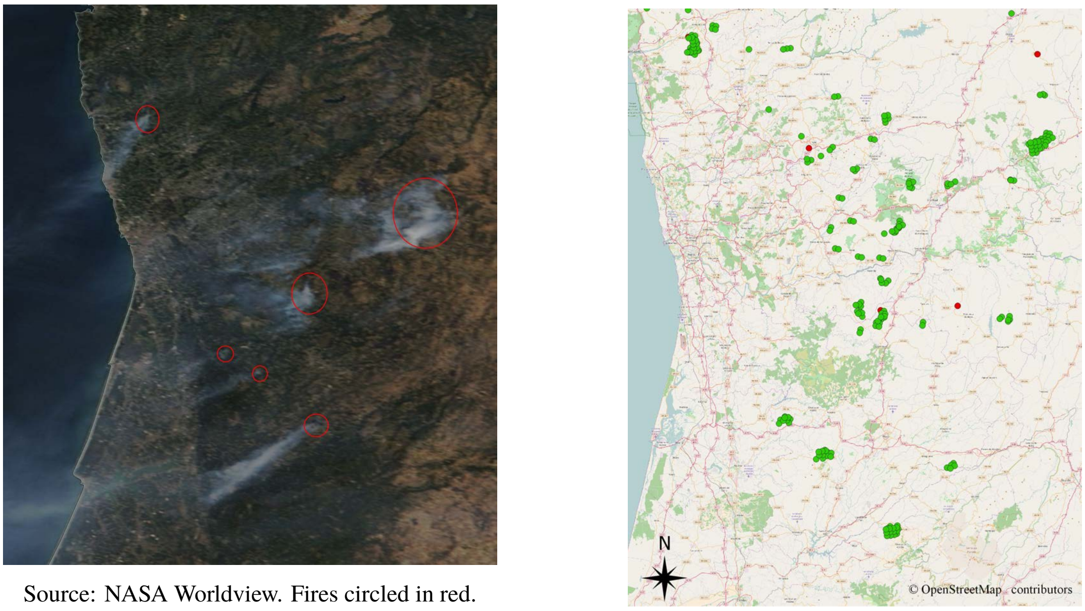
The peaks in the image resemble high frequency of true detections. These true detections are then analyzed on a daily basis to identify the day with the most number of true hits. The code used for this step is given in Appendix E (A.5). For example, there is a peak at the beginning of the second half-year of 2015. Therefore, the true detections from June to September were analyzed followed by the identification of August 9, 2015 as the day with most number of true hits. Then, a satellite image for this day was obtained through historical research to verify if there was really a fire on this day. For verification, the satellite images (surface reflectance images in true colour) were mostly obtained from NASA Worldview [5]. 2 snapshot images for July 20, 2012 and March 18, 2009 were downloaded from Chelys' EOsnap [4].

Some of the detected large fire events are shown in the pictures below. The satellite images are on the left and the results (visualized on QGIS) are on the right side. In the visualizations of the results, green points depict true detections and the red points depict false detections on the particular day. In the satellite images, the visible fires are highlighted using red circles.



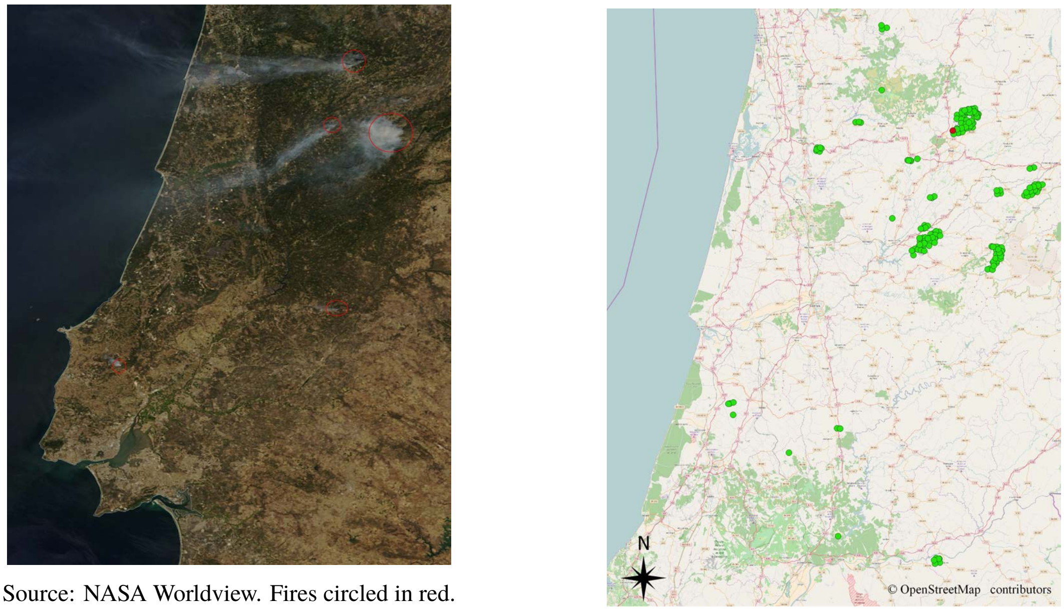
Source: NASA Worldview. Fires circled in red. Fires circled in red.

Figure 33: August 9, 2015.



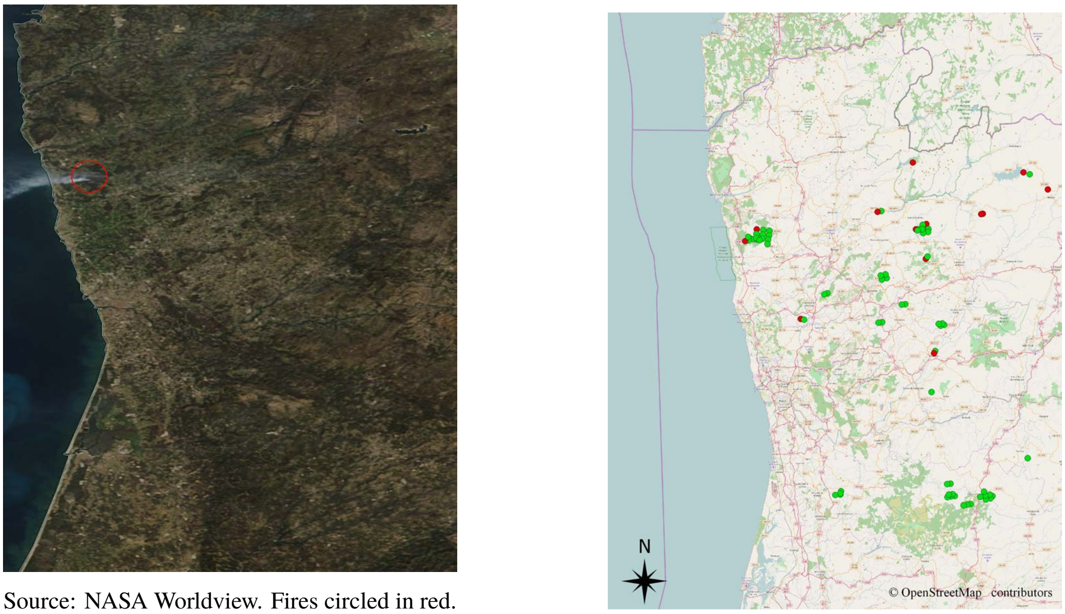
Source: NASA Worldview. Fires circled in red.

Figure 34: August 25, 2013.



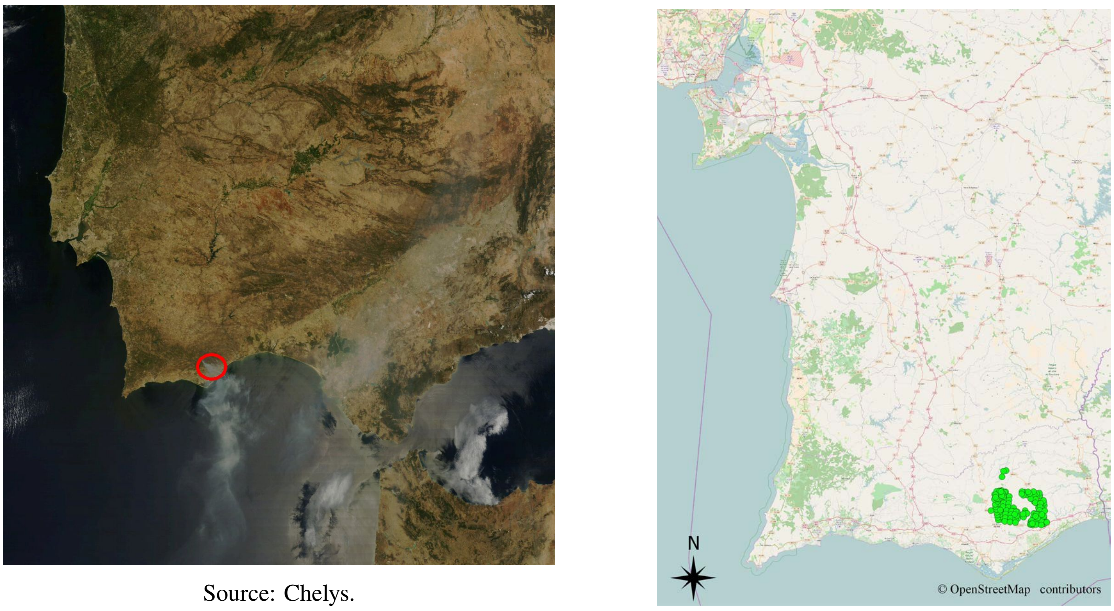
Source: NASA Worldview. Fires circled in red.

Figure 35: September 4, 2012.



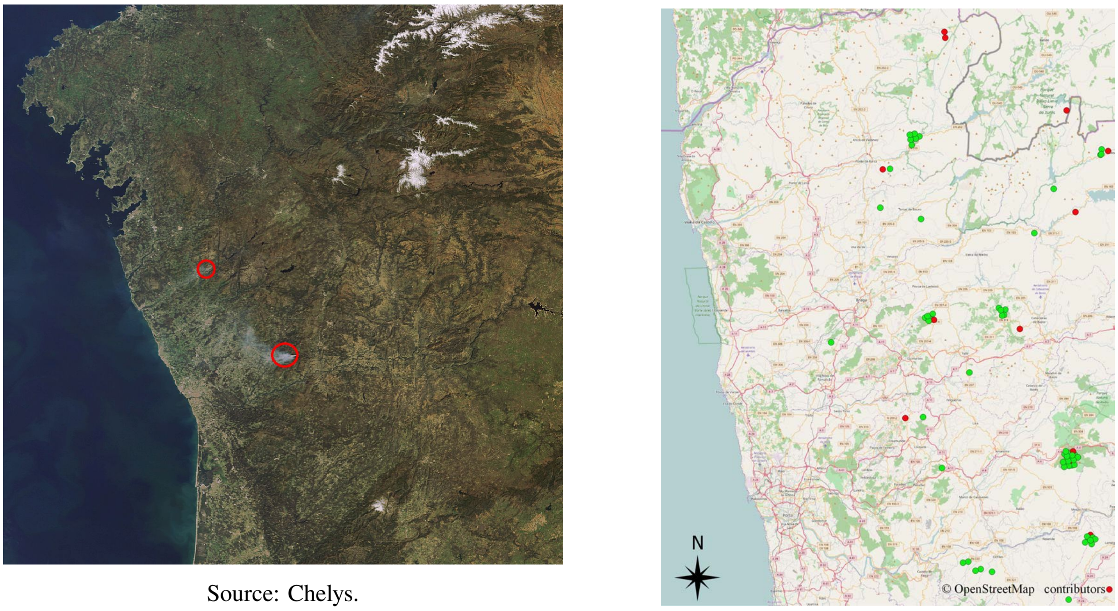
Source: NASA Worldview. Fires circled in red.

Figure 36: March 28, 2012.



Source: Chelys.

Figure 37: July 20, 2012.



Source: Chelys.

Figure 38: March 18, 2009.

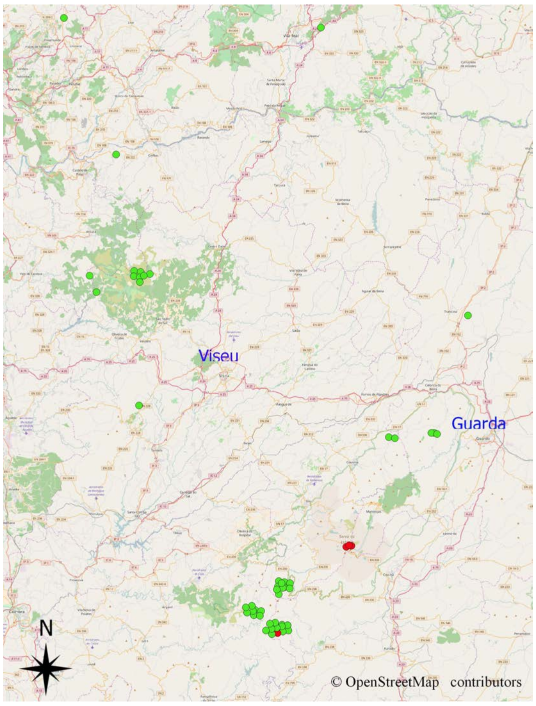


Figure 39: September 10, 2001.

The fires on September 10, 2001 (Figure 39) were verified through a news article that was found on [3]. Fires near the cities of Guarda and Viseu were identified as shown.

5.2 Sensitivity and Specificity Analysis

As already mentioned in sub-chapter 2.4.3, sensitivity, in the perspective of this thesis, is the fraction of correctly classified large fire pixels and specificity is the fraction of correctly classified non-large fire pixels. To judge the effect of each input parameter on the one-class SVM model, a sensitivity and specificity analysis is performed. Each input parameter is removed from the model and predictions are made each time (12 times). The sensitivity and the specificity values are calculated as per the equations 15 and 16, on page 19. The reference for the correct and non-correct predictions is obtained through the model from the previous step which uses all the 12 input parameters. The results are given in Table 9. The sensitivity values for all the parameters was almost equal to 1, but the specificity values were low when the temperature, FRP, land cover and the cluster parameters were removed. For example, without the temperature, the model would not classify any true pixels as false, but would classify almost 50 % of the false pixels as true. This means that the temperature, land cover and ST clustering are the three most influential factors that affect the model, making it reject more pixels than the other parameters. The R code used for this analysis is given in Appendix E (A.5).

	Input_Parameter	Sensitivity	specificity
1	Latitude	1.00	0.95
2	Channel 21 BT	0.99	0.97
3	FRP	0.99	0.87
4	Channel 31 BT	0.98	0.91
5	Temperature	1.00	0.51
6	WindSpeed	0.99	0.90
7	Confidence	0.99	0.96
8	Land Cover	0.99	0.71
9	Elevation	0.99	0.93
10	ST Neighbours (n=1)	1.00	0.90
11	ST Neighbours (n=0)	1.00	0.82
12	Spatial Neighbours	1.00	0.87

Table 9: Sensitivity and Specificity

This means that the model is relatively more responsive to active fires occurring on areas with forest cover, fires occurring in ST clusters and fires coupled with high sir temperatures. This is expected for Portugal as it experiences most of its fires in the summer. The effect of BTs are minimal as shown in the table above. This might be caused due to the fact that the BTs are an intrinsic part of the C6 algorithm to detect active fires (discussed in detail in the next chapter).

6 CONCLUSIONS AND FUTURE WORK

This thesis analyses different factors that affect the MODIS Collection 6 active fire hotspots and combines them using a one-class SVM model to increase the statistical significance of large fire events among the hotspot population. The approach is useful in building a reliable fire database and also in aiding operations related to forest fire management. It basically acts as a filter for the hotspots, narrowing down the possibilities of large fires to a subset of the overall sample set. 79.8 % of the unlabeled fire pixels were predicted as true large fire pixels. These true detections were analysed over time to identify other large fire events in the time period of the study (2000-2016).

The one-class SVM model does not depend on straight-forward conditions where it makes predictions based on individual input parameters. This is one of the biggest advantages of this approach where there is no direct correlation between the input parameters in the data. For example, the results included fires from the months of March, July, August and September. This is an indication that the model does not detect fires only in the summer season with high temperatures. Also, there were fires in southern Portugal, which proves that the predictions were not biased towards the latitude of the fire pixels. That being said, it is also not possible to state the exact reason behind a fire pixel being predicted as a false detection based on its parameter values due to the complex nature of a one-class SVM. Due to the unavailability of labeled samples in the case of false detections, it was not possible to verify the results in terms of false detections.

The model was checked for overfitting, resulting in the exclusion of the month of fire occurrence as an input parameter. The model was not checked for underfitting due to the unavailability of data associated with false detections, as mentioned above. The model was found to be more sensitive to land cover, temperature and the cluster parameters, signifying the clustering nature of the hotspots in forest regions. However, this does not mean that the model rejects all the fire pixels from non-forest areas, those connected to low temperatures or sparsely distributed fire pixels.

It was also observed that both the BTs from Channel 21 and 31 did not have a significant influence on the model. The reason behind this might be that the BTs are already an intrinsic part of the C6 algorithm where they are thresholded in order to filter the fire pixels. In other words, both the labeled and unlabeled fire pixels already belong to a certain interval of the BT measurement scale, resulting in no obvious trend among the labeled samples in particular. On the other hand, FRP being a value calculated by the algorithm and not used for active fire detection as a threshold, did have an impact on the model. FRP is an indirect measurement of the magnitude of a fire. In Tables 2 and 3 on page 33, it can be seen that the labeled samples had a higher FRP on an average when compared to the unlabeled samples whereas the BTs did not show a significant change.

The model could be successfully adapted to other study areas as the model does not use any input parameters that are specific to the study area (Portugal) except for the latitude. This should be taken into consideration for study areas that do not have any relation between forest fires and the latitude, unlike Portugal. The model could also be extended by introducing other external factors such as slope, humidity etc as input parameters and removing the parameters such as BTs. Another development could be increasing the weights of the parameters that the model is more sensitive to, such as land cover or clustering. This might lead to identification of false detections, provided the ground data for false detections is available. Finally, an improved ground research step at the beginning to collect data for known large fires, for instance, by involving the local fire departments' databases, could surely boost the accuracy of this approach.

References

[1] About the Center for Satellite Based Crisis Information (ZKI) . https://www.zki.dlr.de/mission. Accessed: 2017-05-05.

[2] CLC 2012 Copernicus Land Monitoring Service . http://land.copernicus.eu/pan-european/corine-land-cover/clc-2012/view. Accessed: 2016-11-02.

[3] CNN.com - Portugal battles forest fires - September 10, 2001. http://edition.cnn.com/2001/WORLD/europe/09/10/portugal.fire/. Accessed: 2017-04-05.

[4] Earth Snapshot. http://www.eosnap.com/. Accessed: 2017-04-05.

[5] EOSDIS Worldview. https://worldview.earthdata.nasa.gov/. Accessed: 2017-04-05.

[6] For standard data (MCD14ML) extracted from the FIRMS Download Tool: MODIS Active Fire Detections extracted from MCD14ML distributed by NASA FIRMS. Available on-line [https://earthdata.nasa.gov/active-fire-data]. Accessed: 2016-10-14.

[7] Model Fit: Underfitting vs. Overfitting. http://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html/. Accessed: 2017-06-08.

[8] Modis Web. https://modis.gsfc.nasa.gov/about/. Accessed: 2016-10-14.

[9] Nearest Neighbor Index IB Geography . http://www.geoib.com/nearest-neighbor-index.html. Accessed: 2017-05-05.

[10] ZKI MODIS Fire Service for Europe . https://www.zki.dlr.de/services/fire/modis/about. Accessed: 2017-05-05.

[11] Baptista, M. and Carvalho, J. (2002). Fire situation in portugal. *International Forest Fire News*, 27:65–67.

[12] Bowman, D. M., Balch, J. K., Artaxo, P., Bond, W. J., Carlson, J. M., Cochrane, M. A., D Antonio, C. M., DeFries, R. S., Doyle, J. C., Harrison, S. P., et al. (2009). Fire in the earth system. *science*, 324(5926):481–484.

[13] Bucini, G. and Lambin, E. F. (2002). Fire impacts on vegetation in central africa: a remote-sensing-based statistical analysis. *Applied Geography*, 22(1):27–48.

[14] Cochrane, M. A. (2003). Fire science for rainforests. *Nature*, 421(6926):913–919.

[15] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[16] Cortez, P. and Morais, A. d. J. R. (2007). A data mining approach to predict forest fires using meteorological data.

[17] Ferreira, J. E. V., Pinheiro, M. T. S., dos Santos, W. R. S., and da Silva Maia, R. (2016). Graphical representation of chemical periodicity of main elements through boxplot. *Educación química*, 27(3):209–216.

[18] Freitas, A. A. (2000). Understanding the crucial differences between classification and discovery of association rules: a position paper. *AcM sIGKDD Explorations Newsletter*, 2(1):65–69.

[19] Geiß, C., Jilge, M., Lakes, T., and Taubenböck, H. (2016). Estimation of seismic vulnerability levels of urban structures with multisensor remote sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(5):1913–1936.

[20] Giglio, L., Descloitres, J., Justice, C. O., and Kaufman, Y. J. (2003). An enhanced contextual fire detection algorithm for modis. *Remote sensing of environment*, 87(2):273–282.

[21] Giglio, L., Schroeder, W., and Justice, C. O. (2016). The collection 6 modis active fire detection algorithm and fire products. *Remote Sensing of Environment*, 178:31–41.

[22] Goetz, S. J., Fiske, G. J., and Bunn, A. G. (2006). Using satellite time-series data sets to analyze fire disturbance and forest recovery across canada. *Remote Sensing of Environment*, 101(3):352–365.

[23] Hantson, S., Padilla, M., Corti, D., and Chuvieco, E. (2013). Strengths and weaknesses of modis hotspots to characterize global fire occurrence. *Remote Sensing of Environment*, 131:152–159.

[24] Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

[25] Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

[26] JPL, N. (2009). ASTER Global Digital Elevation Model [Data set]. Available on-line [`https://doi.org/10.5067/aster/astgtm.002`]. Accessed: 2016-11-02.

[27] Kraak, M.-J. and Ormeling, F. (2003). *Cartography: Visualization of geospatial data*. Prentice Hall, Harlow, United Kingdom.

[28] Lantz, B. (2013). *Machine learning with R*. Packt Publishing Ltd.

[29] Larose, D. T. (2005). k-nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, pages 90–106.

[30] Lestari, A., Rumantir, G., and Tapper, N. (2016). A spatio-temporal analysis on the forest fire occurrence in central kalimantan, indonesia.

[31] Miyanishi, K. and Johnson, E. A. (2001). *Forest Fires: Behavior and Ecological Effects*. Academic Press.

[32] Müller, D. and Suess, S. (2011). Can the modis active fire hotspots be used to monitor vegetation fires in the lao pdr.

[33] Orozco, C. V., Tonini, M., Conedera, M., and Kanveski, M. (2012). Cluster recognition in spatial-temporal sequences: the case of forest fires. *Geoinformatica*, 16(4):653–673.

[34] Page, S. E., Siegert, F., Rieley, J. O., Boehm, H.-D. V., Jaya, A., and Limin, S. (2002). The amount of carbon released from peat and forest fires in indonesia during 1997. *Nature*, 420(6911):61–65.

[35] Pereira, P. and Turkman, K. (2012). Preliminary analysis of the forest fires in portugal using point processes. *CEAUL, Lisboa*.

[36] Pu, R., Li, Z., Gong, P., Csiszar, I., Fraser, R., Hao, W.-M., Kondragunta, S., and Weng, F. (2007). Development and analysis of a 12-year daily 1-km forest fire dataset across north america from noaa/avhrr data. *Remote Sensing of Environment*, 108(2):198–208.

[37] Roberts, G. and Wooster, M. (2010). Seviri fire radiative power (frp) dataset.

[38] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C., et al. (1999). Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588.

[39] Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.

[40] Stojanova, D., Panov, P., Kobler, A., Džeroski, S., and Taškova, K. (2006). Learning to predict forest fires with different data mining techniques. In *Conference on Data Mining and Data Warehouses (SiKDD 2006), Ljubljana, Slovenia*, pages 255–258.

[41] Tansey, K., Beston, J., Hoscilo, A., Page, S., and Paredes Hernández, C. (2008). Relationship between modis fire hot spot count and burned area in a degraded tropical peat swamp forest in central kalimantan, indonesia. *Journal of Geophysical Research: Atmospheres*, 113(D23).

[42] Tuia, D., Lasaponara, R., Telesca, L., and Kanevski, M. (2008). Emergence of spatio-temporal patterns in forest-fire sequences. *Physica A: Statistical Mechanics and its Applications*, 387(13):3271–3280.

[43] Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

[44] Varela, M. C. (2006). The deep roots of the 2003 forest fires in portugal. *Int. Forest Fire News (IFFN) No*, 34.

[45] Veropoulos, K., Campbell, C., Cristianini, N., et al. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI*, pages 55–60.

[46] Viegas, D. X. (2006). The deep roots of the 2003 forest fires in portugal. *Int. Forest Fire News (IFFN) No*, 34.

[47] Wittenberg, L. and Malkinson, D. (2009). Spatio-temporal perspectives of forest fires regimes in a maturing mediterranean mixed pine landscape. *European Journal of Forest Research*, 128(3):297.

A APPENDICES

A.1 Appendix A: LC Classes Distribution Table

LC ID	LC Class	Frequency	Percentage
33	Transitional woodland-shrub	14830	27.81
21	Moors and heathland	7127	13.36
18	Land principally occupied by agriculture, with significant areas of natural vegetation	4425	8.30
5	Broad-leaved forest	4404	8.26
20	Mixed forest	3552	6.66
9	Coniferous forest	2458	4.61
8	Complex cultivation patterns	2425	4.55
3	Annual crops associated with permanent crops	2377	4.46
31	Sparsely vegetated areas	2092	3.92
22	Natural grasslands	2049	3.84
23	Non-irrigated arable land	1771	3.32
30	Sclerophyllous vegetation	1023	1.92
12	Discontinuous urban fabric	690	1.29
24	Olive groves	630	1.18
35	Vineyards	570	1.07
26	Permanently irrigated land	556	1.04
1	Agro-forestry areas	411	0.77
25	Pastures	355	0.67
6	Burnt areas	304	0.57
17	Industrial or commercial units	239	0.45
15	Fruit trees and berry plantations	221	0.41
27	Rice fields	190	0.36
34	Unclassified	141	0.26
28	Road and rail networks and associated land	105	0.20
19	Mineral extraction sites	102	0.19
36	Water bodies	74	0.14
4	Bare rocks	51	0.10
10	Construction sites	46	0.09
32	Sport and leisure facilities	40	0.08
37	Water courses	24	0.05
2	Airports	18	0.03
13	Dump sites	13	0.02
11	Continuous urban fabric	10	0.02
29	Salines	5	0.01
7	Coastal lagoons	1	0.00
14	Estuaries	1	0.00
16	Green urban areas	1	0.00

Table 10: Percentage distribution of the LC classes for MODIS fire pixels in 2000-2016.

A.2 Appendix B: R Code for Deriving the Input Parameters

To convert the MODIS fire data to a time series object and plot the required graphs in section 3.1.5:

```
#Creating time series object
#grouping by month

#getting the months of each fire
fireMonth_Year <- as.Date(paste(as.numeric(format(shp@data$DATE, "20%y")),as.numeric(format(
      shp@data$DATE, "%m")),"01",sep = "/") ,format = "%Y/%m/%d")

timeSeriesDF <- data.frame(fireMonth_Year)

#creating a vector of all the months in the time period
monthsVector <- seq(min(fireMonth_Year), max(fireMonth_Year), by="months")

#count the frequency of each month
distCount <- plyr ::count(timeSeriesDF)

DfforTS <- data.frame(monthsVector)

#creating a dataframe with fire occurences per month
DfforTS <- dplyr::left_join(DfforTS,distCount,by = c("monthsVector"="fireMonth_Year"))

#assigning the NA values a new class, the months that don't have fires
DfforTS$freq <- car::recode(DfforTS$freq,"NA=0")

#getting the time frame for our timeseries object
minYear <- as.numeric(format(min(DfforTS$monthsVector), "20%y"))
maxYear <- as.numeric(format(max(DfforTS$monthsVector), "20%y"))
minMonth <- as.numeric(format(min(DfforTS$monthsVector), "%m"))
maxMonth <- as.numeric(format(max(DfforTS$monthsVector), "%m"))

#The time series object
timeSeriesFires <- ts(DfforTS$freq, start=c(minYear, minMonth), end=c(maxYear, maxMonth),
      frequency=12)

#Fire Count over time
plot(timeSeriesFires,main = "", xlab="",ylab="", adj = 0.5,
       axes=FALSE,cex.main =3)
axis(1, at=2000:2016, label=paste("", 2000:2016), las=2,cex.lab=1.5, cex.axis=1.5, cex.main
      =1.5, cex.sub=1.5)
axis(2, las =2,  cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
box()

#Box Plot for monthly fire occurence
boxplot(timeSeriesFires~cycle(timeSeriesFires),main = "Monthly frequency of Fires over time",
      xlab="Months",ylab="Fire count")

#creating a new column to store the month of fire occurrence
shp@data$fireMonth <- as.numeric(format(shp@data$DATE, "%m"))
```

To plot the kernel density of the fire distribution in Portugal in 2000-2016 in section 3.1.6:

```

#Converting the fires to a PPP object (point pattern)
#x and y coordinates
x <- coordinates(shp)[,1]
y <- coordinates(shp)[,2]

#coordinates for the bounding box
l1 <- bbox(shp)[1,1]#x
l2 <- bbox(shp)[1,2]#x
l3 <- bbox(shp)[2,1]#y
l4 <- bbox(shp)[2,2]#y

#window of observation
win <- owin(xrange=c(l1,l2), yrange=c(l3,l4))
#class(win)

#converting to ppp
shp_ppp <- ppp(x, y, window=win, marks=shp@data)

#Kernel Density
# Plotting the kernel density of the
K1 <- density(shp_ppp)
plot(K1, main=””)
contour(K1, add=TRUE)
```

To obtain the thresholds in outlier analysis in section 3.2.3:

```
#Obtaining the thresholds
#Temperature
boxplot(FiresExportTable$Temperature,main=”Temperature”,ylab=”Temperature(C)”)$stats

#Elevation
boxplot(FiresExportTable$elevationValues,main=”Elevation”,ylab=”Elevation(m)”)$stats

#Windspeed
boxplot(FiresExportTable$WindSpeed,main=”Wind Speed”,ylab=”Wind Speed(m/s)”)$stats

#Land Cover
boxplot(FiresExportTable$LULC_val,main=”Land Cover”,ylab=”LC class IDs”)$stats

#Confidence
boxplot(FiresExportTable$CONFIDENCE,main=”Confidence”,ylab=”Fire Detection Confidence”)$stats

#FRP
boxplot(FiresExportTable$FRP,main=”FRP”,ylab=”Fire Radiative Power(MW)”)$stats

#channel 21 BT
boxplot(FiresExportTable$BRIGHTNESS,main=”Channel 21 BT”,ylab=”Brightness Temperature(K)”)$
    stats

#channel 31 BT
boxplot(FiresExportTable$BRIGHT_T31,main=”Channel 31 BT”,ylab=”Brightness Temperature(K)”)$
    stats
```

For spatio-temporal analysis in section 3.3.2:

```
##Temporal analysis

#Storing it in a list of dataframes to use it later
listOfTemporalRes <- list()
for(i in 0:10){

  #x meters:: "shp" for all fires, "RealFirePoints" for real fires
  shp_buffer <- gBuffer(RealFirePoints, width = 3058, byid = TRUE)
  #plot(shp_buffer)

  ## For each ID, find IDs within that Buffer
  bak <- shp
  bak@data <- select(bak@data, ID, DATE)
  #overlay the buffer on the target points
  bak_over <- over(shp_buffer, bak, returnList = TRUE)

  res <- data.frame(timeWindow = numeric(), circleID = numeric(), ct_SpTempNeighbors = numeric())

  for(j in 1:length(bak_over)){
    tmp <- bak_over[j]
    #converting to data frame from list
    df <- do.call(rbind.data.frame, tmp)

  #The ID of the buffer origin
    CircleName <- names(tmp)
    df$CircleID <- CircleName

  #Date of the buffer origin
    CircleDate <- df$DATE[which(df$ID == CircleName)]

#Removing the origin, because we don't want it to be considered as a neighbour of itself
    df <- filter(df, ID != CircleName)

    #timedifference, between the date of fire and the date of neighbors, in days
    df$TIMEDIFF <- difftime(df$DATE,CircleDate,units = "days")

    #filtering out the points that are not temporally close enough (0,1,2,3,4..10 days in the past)
    df <- filter(df, TIMEDIFF >= -i & TIMEDIFF <=0 )

    res[j,] <- c(i, as.numeric(CircleName), nrow(df)) #storing the count, number of neighbors
  }

  listOfTemporalRes[[i+1]] <- res
}

  #To store the results
  summary <- data.frame(TimeWindow = numeric(), Mean = numeric(), Median = numeric(), StandardDeviation= numeric(),
                                  Variance = numeric(), Maximum = numeric())
  for(i in 1:11){
    mean <- mean(listOfTemporalRes[[i]][["ct_SpTempNeighbors"]])
    median <- median(listOfTemporalRes[[i]][["ct_SpTempNeighbors"]])
    sd <- sd(listOfTemporalRes[[i]][["ct_SpTempNeighbors"]])

    max <- max(listOfTemporalRes[[i]][["ct_SpTempNeighbors"]])
    variance <- var(listOfTemporalRes[[i]][["ct_SpTempNeighbors"]])

    summary[i,] <-c(i-1,mean,median,sd,variance,max)
  }

  #average number of nearest neighbours plot
```

```
ggplot(summary, aes(TimeWindow), main ="") +
  geom_line(aes(y = Mean ),colour="red") +
  theme(plot.title = element_text(hjust = 0.5,face="bold", size=12),axis.title = element_text(hjust = 0.5,face="bold", size=12),axis.text = element_text(hjust = 0.5, size=12) )+
  labs(x = "Number of Days in the past", y = "Average number of neighbours")+
  scale_x_continuous(breaks = c(0,1,2,3,4,5,6,7,8,9,10))+
  scale_y_continuous(breaks = c(15,16,17,18,19,20,21,22,23,24))
```

A.3 Appendix C: Complete R Code for Creating the Input Table

```r
#Paths
PortugalLandCover <- "LC path"
Elevation <- "Elevation raster path"
temperatureData <- read.table("temperature csv path",sep = ";",header = TRUE)
WindSpeedData <- read.table("windspeed csv path",sep = ";",header = TRUE)

#the point shapefile
path <- "MODIS data shapefile path"
shp <- readOGR(dsn = path,layer = "MODIS layer name")

#create raster lulc
lulcLayer <- raster(PortugalLandCover)

# Extract the raster values underlying the fire points
shp@data$LULC_val <- extract(lulcLayer, shp)

#assigning the NA values a new class
shp@data$LULC_val <- car::recode(shp$LULC_val,"NA=45")

#create raster elevation
elevationLayer <- raster(Elevation)

# Extract the elevation raster values underlying the fire points
shp@data$elevationValues <- extract(elevationLayer, shp)

# Spatio Temporal Neigbourhood Analysis starts here for xkm and n days window
## Copy shp. Remove Data. Add ID
shp@data$ID <- 1:nrow(shp@data)
shp@data$DATE <- as.Date( as.character(shp@data$ACQ_DATE))

#What radius do you want to set here ?? in meters
radius <- 3058
#creating temperature and windspeed column
shp@data$Temperature <- 0
shp@data$WindSpeed <- 0

#3058  buffer
shp_buffer <- gBuffer(shp, width = radius, byid = TRUE)
#plot(shp_buffer)

## For each ID, find IDs within that Buffer
bak <- shp
bak@data <- select(bak@data, ID, DATE, ACQ_TIME)
bak_over <- over(shp_buffer, bak, returnList = TRUE)

#data frame with count of spatial and historical neighbors
res <- data.frame(circleID = numeric(), ct_SpTempNeighbors = numeric(),ct_SpatialNeighbors = numeric()
                  ,ct_SpatialNeighbors_sameDay = numeric())

#2day window
for(j in 1:length(bak_over)){
  # cat(paste0(i, "\n"))
  tmp <- bak_over[j]

  df <- ldply (tmp, data.frame) # http://stackoverflow.com/a/4227483

  CircleName <- names(tmp)
  df$CircleID <- CircleName

  CircleDate <- df$DATE[which(df$ID == CircleName)] # or df$CircleID instead of circle name

  df <- filter(df, ID != CircleName) #filtering out the ID itself, the fire point

  #timedifference, between the date of fire and the date of neighbors, in days
  df$TIMEDIFF <-difftime(df$DATE, CircleDate, units = "days")

  #filtering out the points that are not spatially close enough (2 day window here) n =1
  df_timeFilter <- filter(df, TIMEDIFF >= -1 & TIMEDIFF <=0)

  #filtering out the points that are not on the same day  n=0
  df_timeFilter_sameDay <- filter(df, TIMEDIFF == 0)



  #storing the count, number of spatio temporal neighbors, spatial neighbors
  res[j,] <- c(as.numeric(CircleName), nrow(df_timeFilter), nrow(df), nrow(df_timeFilter_sameDay))
}

#Joining on the basis of ID, which we created in the beginning
shp@data <- dplyr::left_join(shp@data,res,by = c("ID"="circleID"))



######## Monthly Occurrence ############
shp@data$fireMonth <- as.numeric(format(shp@data$DATE, "%m"))


##########Temperature########
colnames(temperatureData) #TEMPERATURE_AVG is the name

#ADDING DATE
temperatureData$Date <- as.Date(as.character(temperatureData$DAY), format='%Y%m%d')

#Add Temperature value for each point
for(j in 1:nrow(shp@data)){

  test <- subset(temperatureData, Date == shp@data[j,]$DATE)

  #making it a spatial point data frame with 4326 projection
  coordinates(test) <- c("LONGITUDE", "LATITUDE")
  proj4string(test)=CRS("+init=epsg:4326") # set it to lat-long
  test = spTransform(test,proj4string(shp))

  #The fire point under observation
  thisPoint <- shp[j,]

  nearestIndex <- apply(gDistance(test, thisPoint, byid=TRUE), 1, which.min)

  #finally, assigning the temperature value, in Celsius
  shp@data$Temperature[j] <- test[nearestIndex,]$TEMPERATURE_AVG

}
#######temperature values added#########

############WindSpeed############
#Renaming Label3 to LanduseClass
colnames(WindSpeedData) #WINDSPEED is the name

#ADDING DATE
WindSpeedData$Date <- as.Date(as.character(temperatureData$DAY), format='%Y%m%d')

#Add Wind speed value for each point
```

```r
for(j in 1:nrow(shp@data)){

  test <- subset(WindSpeedData, Date == shp@data[j,]$DATE)

  #making it a spatial point data frame with 4326 projection
  coordinates(test) <- c(”LONGITUDE”, ”LATITUDE”)
  proj4string(test)=CRS(”+init=epsg:4326”) # set it to lat-long
  test = spTransform(test,proj4string(shp))

  #The fire point under observation
  thisPoint <- shp[j,]

  nearestIndex <- apply(gDistance(test, thisPoint, byid=TRUE), 1, which.min)

  #finally, assigning the windspeed in m/s
  shp@data$WindSpeed[j] <- test[nearestIndex,]$WINDSPEED
}

#Training data
FiresTable <- shp[BigFiresIDs,]

FiresExportTable <- select(FiresTable@data,LATITUDE,LONGITUDE,DATE,BRIGHTNESS,
                           BRIGHT_T31,FRP,Temperature,fireMonth,WindSpeed,CONFIDENCE,
                           LULC_val, elevationValues,ct_SpatialNeighbors,
                           ct_SpatialNeighbors_sameDay,ct_SpTempNeighbors)

#Test data
NonFireTable <- setdiff(shp@data,FiresTable@data )

#for the export, this table also has additional parameters from the modis data
NonFiresExportTable <- select(NonFireTable,LATITUDE,LONGITUDE,DATE,BRIGHTNESS,
                              BRIGHT_T31,FRP,Temperature,fireMonth,WindSpeed,CONFIDENCE,
                              LULC_val, elevationValues,ct_SpatialNeighbors,
                              ct_SpatialNeighbors_sameDay,ct_SpTempNeighbors)
```

A.4 Appendix D: R Code for CV approaches

For Leave-one-out CV in section 4.3.1:

```r
#Number of unique fire dates == number of fire incidents we have
unique_Fire_Dates <- unique(FiresTable$DATE)

#To calculate the accuracy of each model
ListofFireDetections <- list(data.frame())

########Tuning and modeling for each fire event##########
for(i in 1:length(unique_Fire_Dates)){

  print(unique_Fire_Dates[i])#reference
  #current date
  currentDate <- unique_Fire_Dates[i]

  tempShape <- shp
  #removing the current date from the train dataset
  tempTrainData <- FiresExportTable[FiresExportTable$DATE != currentDate ,]

  #Adding the eliminated fire to our test data (the current date fire)
  TestCopy <- setdiff(FiresExportTable,tempTrainData)

  #dummy dependent column :: all True
  tempTrainData$BigFire <- as.logical(”TRUE”)

  #Making x and y columns for the model :: we romove unnecesary columns which we dont need for the model
  #but need to visualize the results later using these columns
  x <- subset(tempTrainData, select = -DATE) #make x variables
  x <- subset(x, select = -ID) #make x variables
  x <- subset(x, select = -BigFire) #make x variables
  x <- subset(x, select = -LONGITUDE) #make x variables
  x <- subset(x, select = -fireMonth)#ADD THIS TO REMOVE THE FIRE MONTH

  y <- tempTrainData$BigFire #make y variable(dependent)

  #Tuning the model, one class SVM, 10 fold cross validation
  tuned <- tune.svm(x=x,y=y,
                    nu =  0.001:1.0,
                    gamma = 10^(-4:0),
                    type='one-classification' )

  model <-svm(x=x, y=y,type='one-classification',
       gamma = tuned$best.parameters$gamma, nu = tuned$best.parameters$nu)

  # test on the left out fire
  Test <- subset(TestCopy, select = -ID )
  Test <- subset(Test, select = -DATE )
  Test <- subset(Test, select = -LONGITUDE )
  Test <- subset(Test, select = -fireMonth )#ADD THIS TO REMOVE THE FIRE MONTH

  pred <- predict(model, Test) #create predictions

  TestCopy$predictions <- pred

  #####Detections for the left out fire event
  ListofFireDetections[[i]] <- TestCopy

}

shp$BigFire <- ”No Fire”
```

```
for(i in 1:length(BigFiresIDs)){
  shp$BigFire[shp$ID == BigFiresIDs[i]] <- ”Big Fire”
}

#For Visualizing all the results
Result_Table <- data.frame(Index = numeric(0), Result_Accuracy = numeric(0),
       DATE = character(0),stringsAsFactors = FALSE)
for( i in 1:length(unique_Fire_Dates)){
  #Current date
  unique_Fire_Dates[i]
  #Test for the first date, use the same template for any date !!
  hh <- ListofFireDetections[[i]]

  #Merging other columns
  hh <- merge(x = hh, y = shp@data[ ,c(”ID”,”BigFire”)], by = ”ID”)

  #Percentage of points that are detected as true and are also labeled as real big fires
  fraction = nrow(hh[hh$predictions==”TRUE” & hh$BigFire == ”Big Fire”,])/
     nrow(hh[hh$BigFire == ”Big Fire”,])
  SuccessPercentage = fraction * 100

  date = as.character(unique_Fire_Dates[i])
  Result_Table[i,]<- c(i,SuccessPercentage,date)
}
#Visualize results
View(Result_Table)
#plot results
plot(Result_Table$Index,Result_Table$Result_Accuracy,xlab='Fire Event Index',ylab='Prediction Accuracy', ylim=c(0, 100))
```

For Hold-out CV in section 4.3.2:

```
copy<- FiresExportTable#Copy before sampling the dataframe
#define fraction of training and test set
bound <- floor((nrow(copy)/4)*2)
  copy <- copy[sample(nrow(copy)), ]           #sample rows

  TempTrainData <- copy[1:bound, ]              #get training set
  TempTestData <- copy[(bound+1):nrow(copy), ]    #get test set
#TempTestData <- copy[1:bound, ]               #for step 2
#  TempTrainData <- copy[(bound+1):nrow(copy), ]    #for step 2

  #Preparing the training data for the model
  TempTrainData$BigFire <- as.logical(”TRUE”)

  x <- subset(TempTrainData, select = -DATE) #make x variables
  x <- subset(x, select = -ID) #make x variables
  x <- subset(x, select = -BigFire) #make x variables
  x <- subset(x, select = -LONGITUDE) #make x variables
  x <- subset(x, select = -fireMonth) #make x variables

  y <- TempTrainData$BigFire #make y variable(dependent)

  #Tuning the model
  tuned <- tune.svm(x=x,y=y,
                    nu =  0.001:1.0,
                    gamma = 10^(-4:0),
                    type='one-classification'   )

  model <-svm(x=x, y=y,type='one-classification',
              gamma = tuned$best.parameters$gamma, nu = tuned$best.parameters$nu)

  Test <- subset(TempTestData, select = -ID )
  Test <- subset(Test, select = -DATE )
  Test <- subset(Test, select = -LONGITUDE )
  Test <- subset(Test, select = -fireMonth )

  pred <- predict(model, Test) #create predictions

  TempTestData$predictions <- pred

  #Percentage of true fires == Prediction Accuracy
  step1 = (nrow(TempTestData[TempTestData$predictions == TRUE,]) / nrow(TempTestData))*100
print(step1)

#step2
  #Percentage of true fires == Prediction Accuracy
 # step2 = (nrow(TempTestData[TempTestData$predictions == TRUE,]) / nrow(TempTestData))*100
#print(step2)

#Plot the results
#Creating spdf
xy <- TempTestData[,c(2,1)]

spdf <- SpatialPointsDataFrame(coords = xy, data = TempTestData,
                               proj4string = CRS("+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"))

spdf <- spTransform(spdf,proj4string(shp))

#the portugal shapefile
Portpath <- "portugal shapefile path"
portugalShape <- readOGR(dsn = Portpath,layer = "shapefile name")
portugalShape <- spTransform(portugalShape,proj4string(shp))
plot(portugalShape)

TrueFires <- spdf[spdf$predictions == "TRUE",]
FalseFires <- spdf[spdf$predictions == "FALSE",]

points(TrueFires,col="green")
points(FalseFires,col="red")
```

A.5 Appendix E: R Code for one-class SVM application

To apply the one-class SVM model on the unlabeled samples in section 5.1:

```r

#Preparing input data for the model
FiresExportTable$BigFire <- as.logical("TRUE")

x <- subset(FiresExportTable, select = -DATE) #make x variables
x <- subset(x, select = -ID) #make x variables
x <- subset(x, select = -BigFire) #make x variables
x <- subset(x, select = -LONGITUDE) #make x variables
x <- subset(x, select = -fireMonth) #make x variables

y <- FiresExportTable$BigFire #make y variable(dependent)

#Tuning the model
tuned <- tune.svm(x=x,y=y,
                  nu =  0.001:1.0,
                  gamma = 10^(-4:0),
                  type='one-classification'   )

#create model
  model <-svm(x=x, y=y,type='one-classification',
              gamma = tuned$best.parameters$gamma, nu = tuned$best.parameters$nu)

#create model with default parameters to check the effect of tuning
#model <-svm(x=x, y=y,type='one-classification')

#Testing on the non fires set
Test <- subset(NonFiresExportTable, select = -ID )
Test <- subset(Test, select = -DATE )
Test <- subset(Test, select = -LONGITUDE )
Test <- subset(Test, select = -fireMonth )

#make predictions
pred <- predict(model, Test) #create predictions

#join the predictions with other columns
Temp_NonFiresExportTable = NonFiresExportTable
Temp_NonFiresExportTable$predictions <- pred

#Percentage of true fires ::Reference 79.8 %
nrow(Temp_NonFiresExportTable[Temp_NonFiresExportTable$predictions == TRUE,]) / nrow(Temp_NonFiresExportTable)

#Plot Results
#Creating spdf
xy <- Temp_NonFiresExportTable[,c(2,1)]
spdf <- SpatialPointsDataFrame(coords = xy, data = Temp_NonFiresExportTable,
                               proj4string = CRS("+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"))
spdf <- spTransform(spdf,proj4string(shp))

truePoints <- spdf[spdf$predictions == "TRUE",]
falsePoints <- spdf[spdf$predictions == "FALSE",]

plot(portugalShape)
points(truePoints,col="green",pch = 20)

plot(portugalShape)
points(falsePoints,col="red",pch = 20)
```

To analyse true detections over short intervals (months or days):

```

#Nunmber of fires in the year xxxx
trueFires <− shp[as.numeric(format(shp$DATE, ”20%y”)) == ”year”,]
nrow(trueFires)#number of true hits in that year

custom_trueFires <− trueFires[trueFires$DATE > ”start date” & trueFires$DATE < ”end date”,]
nrow(custom_trueFires)#number of true hits in this time period

#Creating time series data

#grouping by days
#creating a vector of all the months in the time period
daysVector <− seq(min(custom_trueFires$DATE), max(custom_trueFires$DATE), by=”days”)

timeSeriesDF <− data.frame(custom_trueFires$DATE)

#count the frequency of each month
distCount <− plyr ::count(timeSeriesDF)

DfforTS <− data.frame(daysVector)

#creating a dataframe with fire occurences per day
DfforTS <− dplyr::left_join(DfforTS,distCount,by = c(”daysVector”=”custom_trueFires.DATE”))

#assigning the NA values a new class
DfforTS$freq <− car::recode(DfforTS$freq,”NA=0”)
#Viewing the table to identify the day with the most number of hits
View(DfforTS)
```

For sensitivity and specificity analysis in section 5.2:

```
#Reference from the complete model
TrueFireIDs <− Temp_NonFiresExportTable[Temp_NonFiresExportTable$predictions == TRUE,]$ID
FalseFireIDs <− Temp_NonFiresExportTable[Temp_NonFiresExportTable$predictions == FALSE,]$ID

#labeled Data
#Making input data for the model
FiresExportTable$BigFire <− as.logical(”TRUE”)
x <− subset(FiresExportTable, select = −DATE) #make x variables
x <− subset(x, select = −ID) #make x variables
x <− subset(x, select = −BigFire) #make x variables
x <− subset(x, select = −LONGITUDE) #make x variables
x<− subset(x, select = −fireMonth) #make x variables

y <− FiresExportTable$BigFire #make y variable(dependent)

#Unlabeled data . the non fires set
Test <− subset(NonFiresExportTable, select = −ID )
Test <− subset(Test, select = −DATE )
Test <− subset(Test, select = −LONGITUDE )
Test <− subset(Test, select = −fireMonth )

#Table to store results
Results = data.frame(Input_Parameter = character(), Sensitivity =  numeric(), specificity =
    numeric(), stringsAsFactors = FALSE)
#Removing each column for the analysis
for(i in 1:length(names(x))){
  print(names(x)[i])
  #drop this attribute
  thisAttribute = as.character(names(x)[i])
  drops <− c(thisAttribute)
  temp_x = x[ , !(names(x) %in% drops)]
  temp_test = Test[ , !(names(Test) %in% drops)]

  #Tuning the model
  tuned <- tune.svm(x=temp_x,y=y,
                    nu =  0.001:1.0,
                    gamma = 10^(-4:0),
                    type='one-classification'   )

  model <-svm(x=temp_x,y=y,type='one-classification', gamma = tuned$best.parameters$gamma,
              nu = tuned$best.parameters$nu)

  pred <- predict(model, temp_test) #create predictions

  Temp_NonFiresExportTable = NonFiresExportTable
  Temp_NonFiresExportTable$predictions <- pred

  #True and False IDs for this case
  TempTrueFireIDs <- Temp_NonFiresExportTable[Temp_NonFiresExportTable$predictions == TRUE,]$ID
  TempFalseFireIDs <- Temp_NonFiresExportTable[Temp_NonFiresExportTable$predictions == FALSE,]$ID

  #Values of Confusion Matrix
  truepositives <- sum(table(TempTrueFireIDs[TempTrueFireIDs %in% TrueFireIDs]))
  falsepositives <- sum(table(TempTrueFireIDs[TempTrueFireIDs %in% FalseFireIDs]))
  truenegatives <- sum(table(TempFalseFireIDs[TempFalseFireIDs %in% FalseFireIDs]))
  falsenegatives <- sum(table(TempFalseFireIDs[TempFalseFireIDs %in% TrueFireIDs]))

  #sensitivity
  sensitivity = truepositives / (truepositives + falsenegatives)
  print(sensitivity)

  #specificity
  specificity = truenegatives / (truenegatives + falsepositives)
  print(specificity)

  Results[i,] = c(thisAttribute,sensitivity,specificity)

}
#Rounding off decimals
Results$Sensitivity = format(round(as.numeric(Results$Sensitivity),2), nsmall = 2)
Results$specificity = format(round(as.numeric(Results$specificity),2), nsmall = 2)
```