ANALYSIS OF WILDFIRE OCCURENCE IN AUSTRALIA USING DATA ANALYSIS TECHNIQUES

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

REA BERBERI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

FEBRUARY, 2022

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled **"ANALYSIS OF WILDFIRE OCCURENCE IN AUSTRALIA USING DATA ANALYSIS TECHNIQUES"** and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Dr. Arban Uka
Head of
Department Date:
04, 03, 2022

Examining Committee Members:

Dr. Arban Uka _____

Dr. Julian Hoxha _____

Dr. Igli Hakrama _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Rea Berberi

Signature: _____

ABSTRACT

ANALYSIS OF WILDFIRE OCCURENCE IN AUSTRALIA USING DATA ANALYSIS TECHNIQUES

Rea Berberi

M.Sc., Department of Computer Engineering

Supervisor: Dr. Julian Hoxha

Thousands of human lives are lost every year around the globe, apart from significant damage to property, animal life, etc., due to natural disasters. This project focused on Wildfire prediction. The work has been performed on building a predictive model for wildfires in Australia during the hottest period of the year. Datasets that have been used contain data of fire activities in Australia from 2005 to 2020. The work done for this project is divided into three parts: giving a brief description of algorithms and methods that will be used for predictive models, steps that will be followed for analyzing, preprocessing the data, and finally building the predictive model for Australian wildfires in December 2021.

This project will also cover the topics of big data, deep learning and machine learning. Multiple steps will be followed in order to build the dataset. These steps include collecting an amount of data, using different preprocessing methods and techniques to correct data inconsistencies, and filtering the data used for the following process. Regarding the predictive models, multiple useful algorithms have been included that are being used for data mining, simulation, and testing.

ABSTRAKT

ANALIZA DHE PARASHIKIMI I DUKURISE SE ZJARREVE TE PYJEVE NE AUSTRALI NEPERMJET TEKNIKAVE TE ANALIZES SE TE DHENAVE

Rea Berberi

Master Shkencor, Departamenti i Inxhinierisë Komjuterike

Udhëheqësi: Dr. Julian Hoxha

Mijëra jetë vihen ne rrezik çdo vit në mbarë globin, përveç dëmeve të konsiderueshme në prona, kafshë etj., si pasojë e fatkeqësive natyrore. Ky projekt u fokusua në parashikimin e zjarreve. Gjatë projektit është punuar për ndërtimin e një modeli që ka të bëjë me parashikimin e zjarreve në Australi gjatë periudhës më të nxehtë të vitit. Të dhënat e përdorura përmbajnë të dhëna për aktivitetet e zjarrit në Australi nga viti 2005 deri në vitin 2020. Puna e bërë për këtë projekt është e ndarë në tre pjesë: dhënia e një përshkrimi të shkurtër të algoritmeve dhe metodave që do të përdoren për modelet parashikuese, hapat që duhen ndjekur për analizë, përpunim i të dhënave dhe përfundimisht ndërtimi i modelit parashikues për zjarret australiane në dhjetor te vitit 2021.

Ky projekt do të përfshijë gjithashtu temat e “Big Data”, “Deep Learning” dhe “Machine Learning”. Do të ndiqen disa hapa për të ndërtuar grupin e të dhënave. Këto hapa përfshijnë mbledhjen e një sasie të dhënash, përdorimin e metodave dhe teknikave të ndryshme të përpunimit të të dhënave për të korrigjuar mospërputhjet e tyre dhe filtrimin e të dhënave të përdorura për procesin pasues. Në lidhje me modelet parashikuese, janë duke u përdorur algoritme të shumta të dobishme që vihen ne punë për nxjerrjen e të dhënave, simulimin dhe testimin.

I would like to acknowledge everyone who played a role in my academic accomplishments.

TABLE OF CONTENTS

ABSTRACT iii

ABSTRAKT iv

TABLE OF CONTENTS vi

LIST OF FIGURES ix

CHAPTER I 1

INTRODUCTION 1

1.1 Background 1

1.2 Thesis Objective 2

CHAPTER II 3

AI TECHNOLOGIES FOR DETECTION AND PREDICTION OF WILDFIRE OCCURANCE 3

2.1 AI technologies for prediction models 3

2.1.1 Artificial Neural Network 8

2.1.2 Physical Basis of EO for Wildfire Detection 9

2.1.3 EO-Based Wildfire Burned Area Mapping 10

2.1.4 Optical Based Approaches 10

2.1.5 SAR Based Approaches 14

2.1.6 SAR-Optical Fusion Approaches 17

2.2. Big Data, Data Mining, Machine Learning 19

2.3. Exploring the data 21

2.3.1. Unique value count 21

2.3.2. Frequency Count 22

2.3.3. Variance 22

2.3.4. Histogram 22

2.3.5. Correlation Heat-map between all numeric columns 22
2.3.6. Cluster size Analysis 23
2.3.7. Clustering or Segmentation 23
2.3.8. Outlier overview 23
2.3.9. Outlier analysis for multiple columns 23
2.3.10. Specialized Visualization 23
2.4. Data processing and validation strategies for predictive models 24
2.5. Data mining algorithms 25
2.6. Methods for data interpolation and data extrapolation 26
2.6.1. Data interpolation. 26
2.6.2. Data extrapolation 28
CHAPTER III 29
WILDFIRE PREDICTIONS FOR AUSTRALIA 29
3.1. Datasets 29
3.2. Analyzing the data 30
3.2.1. Historical Wildfires. 30
3.2.2. Historical Weather 33
3.2.3. Historical Vegetation Index 33
3.2.4. Land Class dataset 34
3.2.5. Historical Weather Forecasts dataset 35
3.3. Data Visualization 37
3.4. Data Preprocessing 38
3.4.1. Weather Dataframe 38
3.4.2. Wildfires Dataframe 38
3.4.3. Vegetation Index 38
3.5. Feature Engineering 42
3.6. Test and train data 46

3.7. DCNN Model 45
3.8. Convolutional Layer 45
3.9. ReLU Activation Layer 45
3.10. Pooling Layer 46
3.11. Fully Connected Layer 47
CHAPTER IV 48
FINAL PREDICTION FOR DECEMBER 2021 48
CHAPTER V 53
CONCLUSIONS AND FUTURE WORK 53
5.1. Conclusions 53
5.2. Future Work 54
REFERENCES 55

LIST OF FIGURES

Figure 1. 3-layer Neural Network 9

Figure 2. Proposed methodology for building a dataset [4] 20

Figure 3. Wildfire map during Winter and Summer Seasons 29

Figure 4. Map of 7 regions in Australia 31

Figure 5. Historical Wildfires data 32

Figure 6. Cross-checking, null values 32

Figure 7. Land Class dataset columns 35

Figure 8. Merge of Wildfire Weather with Weather Forecasts data 36

Figure 9. Checking for the reason of null values 36

Figure 10. Average Precipitation Min and Max per Month 37

Figure 11. Slice the dataset over the Region column 38

Figure 12. Necessary imports 40

Figure 13. Estimated Fire Area Regional Subplots 41

Figure 14. Future Engineering: Surface Area 42

Figure 15. Train-test split, Minmax Scaler 44

Figure 16. Windowing dataset 44

Figure 17. Examples of ReLu 46

Figure 18. DCNN example 47

Figure 19. Baseline model..... 49

Figure 20. DCNN Slingshot..... 50

Figure 21. Estimated fire area - prediction..... 51

Figure 22. Fire regions in December 2021..... 52

CHAPTER 1

INTRODUCTION

1.1 Background

During the topic research in the Data Science field , the idea for developing the research and experiments with the main focus in wildfire prediction came up after reviewing a “Call for Code Challenge for Wildfires Predictions” organized by IBM [34]. After doing the necessary studies and reviewing the datasets provided, also environmental protection always having a big interest and impact, it led to the development of this work.

Natural disasters are defined by their recurrence and inevitability however, there are ways in which the dangerous consequences that follow from them can be avoided or reduced. For now, disaster reduction programs are being used to predict the dangers through Artificial Intelligence; these include big data, AI technics, and deep learning algorithms. The aforementioned techniques particularly help with analyzing, visualizing the data, and predicting disasters.

In addition, natural disasters are usually caused by a repetitive natural process that is often unpredictable and thus, they are the ultimate cause of death and economic loss, as well as irreversible damages. To counter this idea, scientists are using machine learning and AI to help in predicting natural disasters (e.g., fires, earthquakes, flooding, and hurricanes) in the hopes of reducing the damage done by these phenomena. So far, the techniques used seem to be effective in managing climate change, which ultimately has its own implications for society and the future.

To have a clearer picture of these implications and how exactly these techniques are being used in real-time, we will discuss some related and past research that has led to the preceding conclusions.

1.2 Thesis Objective

An overview of basic concepts of Artificial Intelligence and Machine learning and the detailed processes of preparing the data are exposed through this document. Data analysis, preprocessing will lead to building the expected future predictive model for natural disasters: in this case wildfires.

The stated goal towards which we aim in the machine learning field is to predict these natural disasters and avoid them in the future or be prepared to minimize the damage. Among others, some of the questions that will be considered during this project are:

1. What are the types of data needed for natural disasters prediction?
2. How can we process the dataset?
3. How can we build a predictive model?
4. How can we improve the predictability of our model?

CHAPTER 2

AI TECHNOLOGIES FOR DETECTION AND PREDICTION OF WILDFIRE OCCURANCE

2.1 AI technologies for prediction models

Every year, wildfires threaten to destroy hundreds of square kilometers of forest due to their destructive power. It's a global catastrophe that's wreaking havoc on the natural world, the economy, and people's health. Drought, wind, lightning, and geography all play essential roles in the incidence and spread of fires, but people are responsible for the majority of them.

The paper reveals that few studies have been done to monitor the problem of wildfires. The unpredictable behavior of wildfires paved the way for this research since, through big data, data mining, and data mining, people will have a clue on the occurrence of wildfires. The nonlinear nature of numerous wildfires is one of the most significant things that make predicting where wildfires shall occur difficultly. Gathering real-time on soil moisture, wind speed, temperature, and other meteorological components are some of the challenges hampering the forecast of this devastating calamity.

To effectively understand this science, people need to consider things such as fuel distribution, intense energy output, weather conditions. In addition, the incorporation of geographical data into wildfire forecast algorithms is an existing new advancement for this field. [4] The frequency and solution of remote sensing are both rising. This makes it potential to use such data in many ways to analyze and predict the outbreak of fires.

It has been discussed that artificial intelligence (AI) is another IT development that researchers can use to develop a great understanding of algorithms allowing trues surveillance of wildfires. Despite AI enabling its users to understand wildfires through

surveillance better, it also assists in finding fire abnormalities, which is a proper traditional approach to fire monitoring.

It is essential to note that training data utilized in both supervised and unsupervised learning is distinct. Robust software instruments are becoming more essential as AI as they are broadly used. As the number of more accessible tools increases, selecting the most appropriate tool is becoming difficult. To manage wildfires, people need to have a history of the outbreak of fires in that particular area illustrated in Chapter three. In the history of the wildfires database of Australia, the project concluded that big data and data mining could be used to analyze this large number of data and develop required solutions. In such as case, test and training data will be used to develop the predictive model, and to monitor the performance metrics of the approach. Validation of the model is important because it gives a final and real-life check of an unknown dataset to confirm the ML algorithm was trained effectively.

Paper concludes by stating that big data machine learning can be employed in assisting to lower the consequences of natural disasters. Big data machine learning has created strategies that help six distinct sections of disaster regulation: early warning damage detection and assessment, monitoring and detection of disaster, durable threats of the disaster, and response planning when a disaster such as fire is detected. It has been proved that wildfires are detrimental and have affected the economy and people living around the places where there is a fire outbreak.

Monitoring wildfires is a major problem. For huge, severe wildfires with unpredictable behavior due to the combination of complex climatic circumstances, intricate geography, and complex fuel structures, it is almost impossible to make predictions about the fire's behavior. There are several factors that contribute to the nonlinear nature of huge wildfires and it is difficult to forecast when and where they will occur since these processes cover a broad range of geographical and temporal dimensions. There are several more challenges to forecasting the spread of wildfires that have already begun, including gathering real-time data on soil moisture, air temperature, wind speed, and other meteorological factors. Addressing all of these factors might assist land and fire management make choices that could save the lives

of firefighters, surrounding inhabitants, and lands, as well as lower the costs of fire extirpation.

Wildfire management may be improved by better understanding the scientific elements that influence fire incidence, behavior, and spread. Fuel distribution, weather conditions, combustion parameters, and intense energy output are some of the most important aspects to consider. Forecasting wildfires might benefit from new, more effective methods and processes as a result of technological advances. Firefighter aid, better logistical service planning, and forest management evaluation of possible risks are just some of the advances, these new technologies are expected to provide. The incorporation of geographical data into wildfire prediction algorithms is an exciting new development for the field. A potential path is the combination of wildfire simulations with atmospheric meteorological models. Researchers are benefiting from this combination in their efforts to better understand the interplay between fire and the atmosphere. Future wildfire monitoring might benefit from the use of Internet of Things wireless sensors and cameras to reach previously inaccessible or dangerous regions.

Academics can use artificial intelligence (AI) to build better algorithms enabling true surveillance of wildfires and finding abnormalities, enhancing the effectiveness of traditional methods of wildfire monitoring. It is possible to detect patterns in huge data using various machine learning methods. On the basis of how they produce predictions from data, these algorithms may be divided into two categories: learning that is both supervised and unsupervised.

The most popular method is SL, or supervised learning [2]. Algorithms in this category are trained by data what conclusions they should make. To train an algorithm using SL, the program's outputs must already be known and the data required to train the algorithm must already be tagged with right responses. Predictive models are built using classification and regression methods. Different approaches are used for predicting categorical and continuous answers. Algorithms like linear and logistic regression, support vector machines, and artificial neural networks are all included in SL.

Unsupervised Learning (UL) approaches, on the other hand, are often used in data exploration since they are used to uncover trends in datasets from unorganized input data. Unsupervised clustering is the most prevalent method of learning. The dataset is divided into equal-sized clusters, and then each cluster is analyzed separately. Clustering by k-means and association rules are two examples of unsupervised learning algorithms.

It's important to note that the training data utilized in supervised vs unsupervised learning is fundamentally different. Sustained and labelled examples of input data are utilized in the learning process whereas unseen examples are used in an unsupervised learning process. Powerful software tools are becoming more important as Artificial Intelligence becomes more widely used. The selection of the most appropriate tool is becoming more complex as the number of accessible tools increases. Oracle, SAS, SPSS, Microsoft, SQL Server, Teradata, and TIBCO are just a few of the prominent firms that have incorporated. The collection, preprocessing, and analysis of data by artificial intelligence systems yields valuable information. In addition to catastrophe monitoring, agriculture, and water management, these technologies have the ability to solve a wide range of issues. Three dimensions of Big Data can characterize wildfires: Variety, Volume and Velocity [4]:

1) **Volume**: Every day, enormous amounts of data are created. Large amounts of data are now measured in Extrabytes rather than in terabytes. Managing massive amounts of data from many sources is essential for keeping tabs on wildfires. NDVI, LAI, and LST are only some of the useful data that produces remote sensing for wildfires (e.g., from satellites like Terra, Landsat, and Sentinel). For wildfire monitoring, in-situ weather stations may also provide significant information. There is a wealth of information that can be gleaned from sensors that may be placed in forests to capture real-time data. Wildfire monitoring might benefit greatly from all of the data sources listed above.

2) **Variety**: When it comes to monitoring wildfires, satellite photos may provide a broad range of information, such as multi-sources, multi-temporal (acquired at different times), and multi-resolution data. Stations that measure the weather produce a variety of data types (such as air temperature and humidity) as well as measurements

of solar radiation, wind speed, and direction. Wildfires may be predicted and spread more accurately with the use of this information.

3) **Velocity**: Big Data is growing at a rapid pace, with daily volumes approaching 4 terabytes. In the world of big data, there is no limit to the amount of data that can be generated in an instant. Big Data velocity encompasses both rapid data creation and excellent data processing and analysis efficiency. That is to say, to complete a job, the data must be evaluated over a period of time that is both actual and realistic. Instantaneous data processing may save tens of thousands of lives in the setting of wildfires.

Big data is mostly derived via remote sensing. Fast-developing technology has the benefits of a constant, repeatable, large-area coverage as well as the ability to quickly supply a tremendous volume of data from faraway places.

Artificial Intelligence may be used to control it. It may be difficult to identify the presence of fires due to technical or natural limits, which might lead to erroneous statistics. In addition, there is a dearth of data when it comes to managing wildfires. Artificial Intelligence and high-quality data are often relied upon by scientists to develop models that provide significant learning and outcomes. The problem is that AI can't work with very little data. To build a knowledge base and find new items, machine learning algorithms need a large amount of high-quality data.

It is difficult for researchers to aggregate data from numerous sources, and they are often constrained by incomplete or missing data. Even though wildfire researchers face many problems, new methodologies and tactics are helping them overcome this. In order to combat wildfires, a number of AI-based technologies have been created. Neural Networks (NN) have been used to anticipate human-caused wildfires. Wildfire false alarms were reduced by 90% using a combination of infrared scanners and NN. Using satellite pictures and a spatial clustering method (FAST Cid), researchers discovered wildfires. Based on Kaler [17], fires in North America were detected with 75% accuracy at the 1.1-km pixel level by a Support Vector Machine (SVM) in 2005 using satellite photos supplied into the SVM. Wildfires in Slovenian woods were predicted using satellite-based and meteorological data by utilizing

numerous data mining approaches such as Logistic Regression, Decision Trees (DT), and Random Forest (RF).

Geographic data, meteorological data, and multi-temporal make up the three categories of data. In order to create wildfire prediction models, the researchers also required positive and negative samples of fire incidence. These samples were from places where wildfires had occurred in the past, and the date and time were recorded.

Negative samples in the database are represented by a variable frequency with odd timestamps and locations. With an overall accuracy rate of 80%, the best model was created by Bagging of decision trees, which had the greatest prediction accuracy, kappa statistics, and precision out of all the methods tested. In comparison to the high-resolution satellite photos, the expense of collecting the data is little. This study aimed to use five alternative data mining techniques, including Support Vector Machines, and four feature choices to estimate the burnt area in northeastern Portugal. For the suggested approach to work, it needed to be able to anticipate minor fires, which account for the vast majority of flames. However, this method has a poor degree of accuracy when it comes to predicting big fires.

2.1.1 Artificial Neural Network

The term "Artificial Neural Network" refers to a framework of machine learning algorithms that is loosely influenced by the structure and activity of the human brain. The system is composed of interconnected nodes, sometimes referred to as artificial neurons. Edges, or connections between nodes, act similarly to a synapse in a human brain in that they convey a signal from one node to another. When a signal is received by an artificial neuron, it is processed before being sent to other nodes.

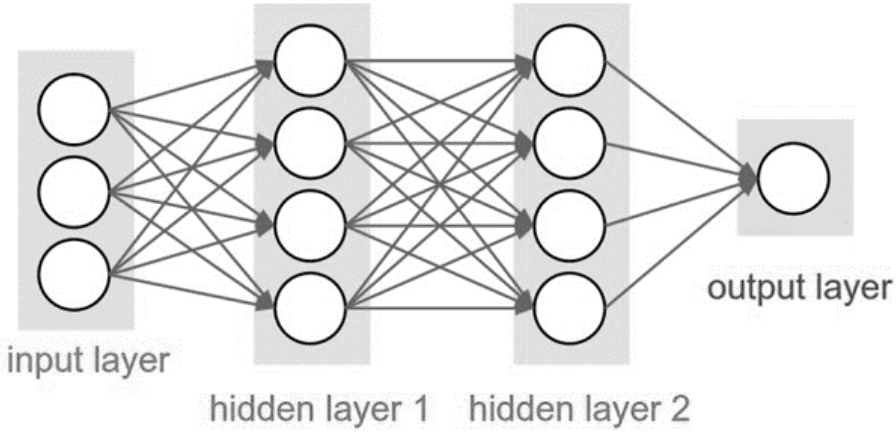


Figure 1. 3-layer Neural Network

2.1.2 Physical Basis of EO for Wildfire Detection

Due to the variety of fire types and behaviors, as well as the time interval between fire extinguishment and image acquisition, the effects of wildfires on vegetation can be rather diverse based on Chuvieco et al. [36]. According to the vegetation layer in which it burns, fire is categorized into three types: surface fire, crown fire, and subsurface fire. When the tree cover is extensive, it is difficult to identify surface flames using EO measurements; crown fires are easy to detect, but only thermal sensors can detect subsurface fires. In terms of fire behavior, the more intense the fire, the more complete the combustion, and the more prominent the ash and charcoal are in terms of spectral contribution in comparison to green vegetation. Due to the fact that the longevity of post-fire signals varies between climate zones, it is crucial to understand the temporal gap between fire extinction and image acquisition.

Wildfires can remove vegetation, expose soil, and alter the moisture content of the soil and vegetation, all of which can affect optical reflectance and radar backscatter. The following section provides an overview of the optical spectra and SAR backscatter variations generated by wildfire, as well as their significance.

2.1.3 EO-Based Wildfire Burned Area Mapping

Methods for detecting wildfires have advanced significantly over the last two decades, from early single-date visual analysis through dimensionality reduction, multi-temporal change detection, time series analysis, classification, and regression algorithms. Three significant hurdles remain for algorithms for mapping burned areas and estimating burn intensity using EO data:

1) The relationship between explanatory variables (spectral reflectance) and the response variable (burned/unburned or burn severity) varies according to spatial location and vegetation type/conditions;

2) A scarcity of annotated training data on a variety of land cover types and geographical regions around the world, which severely limits model generalization;

3) A data imbalance problem, which affects both the proportion of burned and unburned areas, as well as vegetation types, topography, and climate zones.

Recent improvements are discussed and classified according to the relevant EO data types, including optical-based, SAR-based, and SAR-Optical fusion techniques.

2.1.4 Optical Based Approaches

The dynamic auto-encoder consists of two stages:

(1) System identification, during which the state estimate is determined by maximizing the likelihood of its one-step (one week) future forecast, and

(2) Fire prediction, during which the ground truth is predicted after four time steps (one month).

The dynamic auto-encoder employed in this work is fed a multidimensional time series extracted from the handpicked 11-channel data set. The encoder transforms the input and passes it along to the recurrent neural network (RNN). The decoder employed the hidden state of the RNN to reconstruct the ground truth input data and estimate the likelihood of new fire spots at following time steps. To determine the accuracy of the predictions at specified time steps, an auto-in-dependent encoder's loss function was tuned. Additionally, the RNN state estimate accurately captures compressed data from the history of available and missing observations, as well as unmeasured variables. Due to the real-time advantage of the dynamic auto-recursive encoder's updating, we can forecast the firegrid map online.

The bulk of global burn area products are based on optical sensors with extremely high temporal resolution (one or more photographs per day) but poor spatial resolution (250m), such as MODIS MDC64A1, FireCCI50, and NASA's MCD45A1. Global burn area mapping algorithms were intended to be resilient and spatially adaptive in order to deal with the broad variety of wildfire conditions, vegetation types, and topography found throughout the world. In recent years, the most frequently utilized methodologies for global burn area mapping have been those that are locally customized and physical-based. With recent developments in cloud computing capacity and open access to Landsat and Sentinel-2 data, the emphasis has switched to generating regional or national products using optical data with a medium resolution. The 2008 public release of Landsat archival data ushered in a new age of using medium resolution data to retrieve vegetation changes on a regional or even global scale, hence enhancing long-term time series detection techniques.

Time series segmentation and time series decomposition are the two most often utilized strategies for detecting time series forest disturbances. Kennedy et al. suggested a method for detecting and categorizing forest disturbances based on dense time series pictures from Landsat TM and ETM+ that incorporates all time series photos and employs an idealized temporal trajectory of spectral bands to detect and characterize changes [37]. Two more approaches, the Vegetation Change Tracker (VCT) and the Landsat-based detection of Trends in Disturbance and Recovery (LandTrendr), were developed based on the trajectory segmentation strategy [37].

Recognizing that change is more than a contrast between two points in time, but a continuous process occurring at both fast and slow rates across landscapes, both VCT and LandTrendr segment annual time series of spectral responses into piecewise segments, detect changes between segments, and exploit segment characteristics to delineate forest disturbances [36]. Given that changes in plant ecosystems occur as a result of phenological, gradual, and abrupt changes, Verbesselt et al. proposed the BFAST (Breaks For Additive Seasonal and Trend) method for decomposing EO time series into trend, seasonal, and residual components for the purpose of detecting forest changes [36]. Verbesselt et al. developed a multi-purpose near-real-time disturbance monitoring approach based on the BFAST season-trend model. This approach analyzes time series and automatically identifies and models the stable history portion of the time series in order to detect disturbances within newly acquired data [37]. Similarly, Zhu et al. introduced a Continuous Change Detection and Classification (CCDC) algorithm for segmenting dense Landsat time series into seasonality and long-term patterns and classifying change as a departure from those trends. While both time series segmentation and decomposition have the capacity to detect changes in land cover, additional attribution is necessary to describe certain forms of change, such as thermal anomaly for the detection of burn areas. Numerous change detection techniques have been developed and widely applied in optical-based land cover change detection over the last few decades [37]; including burn area mapping.

The bulk of existing algorithms to change detection were developed to detect bi-temporal spectral changes in nature, which means they can only be used to examine an image pair over a single geographical area at a time [37]. Additionally, satellite time series can be separated into image pairs for the purpose of utilizing bi-temporal change detection algorithms. On the other hand, burn area mapping algorithms seek to detect spectral changes generated by wildfire that are suitable to optical wavelength remote sensing and vary spatially and temporally based on the pre- and post-fire vegetation structure and land cover condition (Boschetti et al.). As a result, the majority of burn area mapping applications used an absolute change detection technique to subtract a post-fire image from a pre-fire image in order to derive differentiated spectral or indices, such as differentiated NDVI (dNDVI) in NIR-Red space and differentiated NBR (dNBR) in NIR-SWIR space. Miller et al. developed a relative dierenced NBR

(RdNBR) by dividing dNBR by the square root of the pre-fire NBR in order to reduce pre-fire vegetation's biasing effects on burn severity. Massetti et al. proposed the Vegetation Structure Perpendicular Index (VSPI), which was defined as the deviation from a linear regression between two SWIR bands centered at 1.6m and 2.2m in a time series. The VSPI index exhibited less interannual variability and a stronger post-wildfire detection of disturbance over a longer time period than the NBR and NDVI indexes, implying a more comparable measurement of wildfire effects, maybe closer to dNBR and dNDVI.

Over the years, a variety of machine learning techniques for identifying burn areas from differentiated spectral bands or indices have been developed, including the Bayesian classifier, Random Forest (RF), Decision Trees (DT), Region Growing, Support Vector Machine (SVM), and deep neural networks (DNN). The majority of these approaches are data-driven, with performance being decided by the quality and volume of available data. Boschetti et al. presented a spectral-rule-based DT for Landsat 30m data in order to identify candidate burned areas that were then maintained or deleted based on comparisons to concurrent MODIS active fire detection. Crowley et al. used Bayesian Updating of Land Cover (BULC) to merge burn area data from numerous optical sources, including Landsat-8, Sentinel-2, and. Roy et al. employed optical Landsat-8 and Sentinel2 data to map burned regions using a random forest change regression trained on synthetic data constructed from laboratory and field spectra and a spectral model of fire effects on reflectance. Knopp et al. employed the U-Net to semantically separate burned areas using Sentinel-2 data; the result is a homogenous burned area mask that completely fills in small unaaced. Pinto et al. presented a BA-Net for burned area mapping that combines the Long-Short Term Memory (LSTM) with the U-Net and produced competitive results using VIIRS 750m data. Machine learning approaches based on optical data have advanced dramatically in terms of development, accuracy, and computational efficiency when used to map burned areas and burn severity. On the other hand, regional or global medium resolution burned area/burn severity products are still unavailable. Developing globally applicable models from scant annotated data and a vast amount of unlabeled optical data continues to be a significant difficulty. Due to cloud cover, smoke, and night, the ability for monitoring wildfires with medium resolution optical data remains

limited. This thesis is primarily concerned with the use of Sentinel-1 C-Band SAR data for mapping wildfire progression.

2.1.5 SAR Based Approaches

SAR data has regularly been utilized to map burned areas over tropical regions with constant cloud cover [37] or at high latitudes where optical observations were impeded by low sun angles. SAR data has the ability to improve temporal resolution for following additional progressions and mapping burned areas of current wildfires in various climate zones or geographic places where optical observation may be limited by dense smoke cover. While SAR has been used to monitor forest change on a wide scale, SAR signals over vegetation changes are usually more confusing to interpret than optical observations. In SAR-based investigations, the backscatter coefficient, interferometric coherence, or polarimetric properties (as determined by polarimetric decomposition techniques) are employed to quantify fire impacts. However, both sensor and scene parameters may have an effect on the SAR backscattering process investigated in these investigations assessed burn severity using X-, C-, and L-band co/cross-polarization SAR backscatter, as well as the effect of local topography and meteorological circumstances on SAR backscatter response. The sensitivity of radar backscatter coecients to burn severity rises as the radar wavelength increases, and the local incidence angle has a significant effect on backscatter coecients from burned areas at all wavelengths and polarizations.

Additionally, the relationship between burn severity and interferometric coherence was examined, and it was reported that the strongest correlation between coherence and burn severity was observed in images taken under steady, dry environmental circumstances. Coherence grew linearly from unburned to extensively burned forest, from 0.5 to around 0.8, and saturation developed at high dNBR values. When LIA was included, the determination coefficients increased from 0.6 to 0.9 at the X- and C-bands, although LIA had a smaller effect on the strength of the correlation between burn severity and L-band. Numerous radar indices, analogous to optical spectral indices, have been developed to characterize vegetation dynamics, including

the Radar Vegetation Index (RVI), the ratio VH/VV (cross ratio, CR), and the Radar Forest Degradation Index (RFDI).[37] Kim et al. studied the link between vegetation water content (VWC) and X-, C-, and L-band RVI throughout the growth cycle and discovered that L-band RVI showed a larger correlation with VWC than C-band RVI. Veloso et al. analyzed Sentinel-1 and NDVI time series for wheat, oilseed-rap, corn, soybean, and sunflower, demonstrating a strong correlation between SAR and NDVI, as well as a strong correlation between the VH/VV ratio and in-situ biomass for barley and corn.

Belenguer-Plomer et al. studied the applicability of eight temporal indices of bi-temporal radar backscatter coecients for burned area mapping using an RF classifier, which has the capability to quantify and analyze temporal indices' significance. This study showed that algorithms for mapping burned areas should use different radar indices for different vegetation types, and despite their widespread use in fire monitoring using SAR data, log-ratio indices were not shown to be more important than simple ratios. Engelbrecht et al. proposed a Normalised Dierence — Angle Index (ND–I) for mapping burn areas using C-band data, in which polarimetric decompositions (H-A– decomposition for quad-pol RADARSAT-2 and H— decomposition for dual-pol Sentinel-1) were used to derive —angles from pre- and post-fire SAR data, and ND–I was calculated to identify the burn areas. Frequently, algorithms for mapping burned areas using radar pictures are based on temporal discrepancies between pre- and post-fire SAR data induced by fire occurrences. In general, the burning of scattering materials results in a decrease in the backscatter coefficient in burned areas. However, when rainfall occurs following a fire occurrence, the reverse trend may be observed, as the elimination of vegetation results in greater soil surface scattering, particularly for co-polarized waves. Belenguer-Plomer et al. employed the Reed-Xiaoli Detector (RXD) to identify anomalous variations in the SAR backscatter coecient, which they then compared to MODIS and VIIRS thermal hotspot detections.

Nartubs et al. examined changes in the L-band PALSAR signal and tri-dimensional polarimetric responses to various types of fire disturbance in the northern Amazon, and discovered that the polarimetric L-band PALSAR data were sensitive to

changes in forest structure and Above Ground Biomass (AGB) caused by forest fire. No one PALSAR feature, however, was capable of discriminating between intermediate levels of fire degreadation. Belenguer-Plomer et al. investigated the temporal correlation of Sentinel-1 SAR backscatter coefficient with RF over burned areas in Mediterranean ecosystems, concluding that fire severity and water content (in soil or vegetation) were the most significant factors affecting the temporal correlation across all land cover classes except herbaceous, and that, in general, the backscatter decrease in burned areas was significant. SAR time series were utilized in a few studies to monitor forest disturbances. Several studies have been published on this topic.

[36] Dong et al. investigated the feasibility of using C- and L-band SAR time series to monitor changes in Indonesian plantation and natural forests, concluding that while both VV and VH are important for distinguishing forest types at C-band, HV carries the majority of the relevant information at L-band, and L-band is significantly more sensitive to forest changes than C-band. Reiche et al. used a dense Sentinel-1 time series in conjunction with VIIRS-based active fire alerts to assess tropical forest loss caused by fire. Pre-fire time series were segmented into training and monitoring periods, and a harmonic model was fitted to account for forest seasonality across time and then used to deseasonalize training period time series observations. The deseasonalized time series were used to construct pixel-specific forest and non-forest distributions, which were then utilized to parameterize a probabilistic technique for identifying near-real-time forest cover reduction during the monitoring period. Zhou et al. used long-term SAR backscatter time series to monitor post-fire plant recovery in the Tundra environment. C-band SAR time series revealed that burned regions required five years to restore to pre-fire levels, which is longer than the three years suggested by optical NDVI observations. SAR's utility in mapping burned areas has been well established, and the literature has also examined the link between SAR backscatter and optical reflectance or indices. However, there is still a research gap about how to apply deep learning to map the progression of wildfires in near real time when there is no external data or labels available for the area under consideration. Additionally, it is unknown how the performance of deep learning models may differ depending on whether they are trained with SAR-based (noisy) labels or optical-based

labels, given the disparities in sensitivity of SAR and optical sensors to various levels of burn severity.

2.1.6 SAR-Optical Fusion Approaches

Rather of relying exclusively on optical or radar sensors, merging multiple EO sensors has demonstrated significant potential for improving burn pixel discrimination and minimizing both omission and commission errors. The most frequently utilized synergistic technique combines thermal anomalies (active fire alarms) with changes in optical reflectance or SAR backscatter, as explained under optical/SAR-based approaches. This section discusses the use of SAR and optical data in conjunction with one another to monitor forest disturbances. While multi-sensor approaches combining SAR and optical sensors have demonstrated significant improvements in land cover mapping accuracy , the use of SAR-optical time series for detecting forest changes has remained relatively limited to date, leaving a lot of untapped potential. Approaches based on SAR-optical fusion have the potential to enhance observation frequency, hence reducing the time required to detect change events and increasing the accuracy of forest disturbance mapping. Few researchers have examined the application of SAR-optical time series to forest disturbance monitoring. Reiche et al. presented a pixel-based Multi-sensor Time-series Correlation and Fusion (MulTiFuse) approach for detecting deforestation in tropical locations where cloud cover limits optical time series monitoring. MultiFuse models the link between SAR and optical univariate time series using an optimum weighted correlation, and the optimized regression model is utilized to predict and fuse SAR-optical time series. In (Reiche et al., 2015a), a Bayesian approach was used to combine L-band PALSAR and Landsat time series for near real-time deforestation detection. The conditional probability of deforestation (CPD) was computed using Bayesian updating to indicate a deforestation event, and subsequent observations can be used to update the CPD to confirm or reject a candidate deforestation event. By merging dense Sentinel-1 time series with Landsat and ALOS-2 PALSAR-2 data, this strategy was further researched in order to improve near real-time deforestation monitoring in tropical dry forests. Hirschmugl et al. used a combination of Sentinel-1 SAR and optical Sentinel-2/Landsat-8 time series to map

forest disturbance, first calculating an initial forest/non-forest mask based on optical time series and then updating it with detected disturbance from SAR-optical stacks fused with Bayes' theorem, which achieved the highest detection accuracy compared to SAR or optical alone in a complex tropical forest site in Peru.

BelenguerPlomer et al. (2021) investigated the optimal CNN configuration for mapping burned areas based on land cover class and discovered that integrating SAR and optical data can achieve higher accuracy than either SAR or optical data alone, with the highest and lowest accuracies achieved over forest and grasslands, respectively. Due to the major disparities in imaging process and geometry, directly stacking SAR backscatter and optical reflectance/indices may not be the optimal method for fusing multi-source data. A successful model for continuous wildfire advancement mapping typically requires high-quality labels, yet it is well established that SAR is frequently impacted by significant background noise, making it difficult to derive appropriate labels merely from SAR data. It is yet unknown how optical data can be combined to enhance SAR-based labels and hence provide more trustworthy supervision. However, revisiting/reusing all previous training data is time intensive and computationally expensive; how can we continuously refine current models without relying on earlier training data and resulting in a major loss of progression mapping accuracy? Additionally, given the scarcity of labeled wildfire datasets, self-supervised learning of usable representations from unlabeled SAR-Optical data pairings could be an attractive direction for large-scale wildfire detection and monitoring using machine learning/deep learning. Additional study is needed to integrate the benefits of medium resolution SAR and optical time series, as well as to develop a priori spatio-temporal consistency for monitoring wildfire-induced environmental changes.

2.2 Big Data, Data Mining, Machine Learning

Numerous new Artificial Intelligence technologies have been generally applied for developing natural disaster prediction models. These include Big Data, Machine Learning, and Data Mining. These frameworks are discussed in detail below:

[32] Big Data is a group of huge unstructured datasets that are consistently growing past the capacity of basic data tools. These sets are often being used to store, manage, and analyze the data. However, big data is not defined only by the size of the data but it also involves techniques, frameworks, and tools that work efficiently and effectively with the data.

Data Mining includes extracting needed information from large, unstructured data; otherwise known as Big Data. This is done through machine learning, statistical analysis and database technology; which are being combined to build relationships between datasets and get the required information.

Algorithms based on computer-assisted learning are included in the field of machine learning. Algorithms are used to extract meaningful information from the data that is supplied. [3] It is also important to note that in machine learning, there is just one rule: it constructs algorithms that receive input data and apply statistical analysis to forecast new values. Furthermore, machine learning uses a variety of algorithms in order to discover new ways that lead to significant insights. These models may be divided into two categories depending on their approach to interpreting data and making predictions: learning that is both supervised and unsupervised.

The most common method of instruction is one in which students are closely monitored as they go through their coursework called supervised learning. It's up to algorithms to figure out from the data what conclusions they should draw. Supervised Learning requires labeled training data. It uses regression and classification techniques to develop the predictive models; in which the former is used to predict dimensional/continuous responses and the latter is used to predict categorical responses. Supervised Learning includes algorithms such as linear and logistic regression, Artificial Neural Networks, and Support Vector Machines.

On the other hand, **Unsupervised Learning** is used with unlabeled input data. Clustering is the primary UL method, and it entails separating the dataset into several groups. K-means clustering and supervised learning-like principles are part of its algorithmic framework. As a result, the primary distinction between supervised and UL is the kind of data utilized for training.

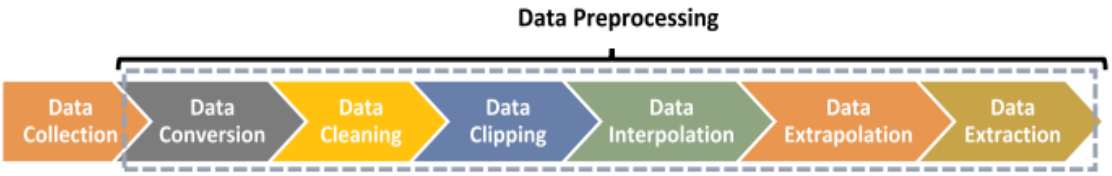


Figure 2. Proposed methodology for building a dataset [4]

Big Data's primary goal is to increase society's long-term ability to prevent calamities and enhance resilience. Preventing or mitigating, monitoring or predicting, and responding or recovering are all part of the action cycle. Referring to paper [3], the authors underwent two main steps which included: (1) Obtaining information, and (2) Deep learning pipeline: Transfer learning for feature extraction and multi-layer perceptron models (MLPs)

The term "transfer learning" refers to the process through which a person applies the skills and information gained from addressing one issue to another, even though the two problems are unrelated. Deep learning algorithms are also constrained in their performance by the number of their training sets. No matter how you slice it, deep learning benefits from a big training set because to transfer learning.

Learning is also influenced by the idea of multi-layered perception. According to this study, the authors built two multilayer perceptron networks after extracting each model's feature vectors in order to categorize the photos by time, period, and urgency for those that had been tagged. An additional step was to link up the photos and labels for each model before randomly dividing the data into two sets: one for training and the other for validation.

2.3 Exploring the data

The first step in data analysis would be data exploration; this is done through exploring and visualizing the data in order to discover insights that stem from the beginning or identify areas/patterns that are to be investigated further. By using interactive dashboards and point-and-click data exploration, the customer or user can understand the bigger picture faster and better, which can consequently lead to insights faster. Some of the methods used for data exploration:

2.3.1 Unique value count

To better understand the data we are working with, it is important to check the number of unique values.

2.3.2 Frequency Count

This method is useful to find how frequently values occur in a column. This algorithm determines the number of times your program is executed, which is dependent on the loop utilized in the program.

2.3.3 Variance

In order to perform a better analysis to numeric values, checking at variance, min, max would be so useful. Variance lets you understand more about values spreading.

2.3.4 Histogram

One of the most useful tools for data scientists is the histogram. It's a handy tool for learning about a dataset's full range of data. It also reveals whether or not data is skewed.

2.3.5 Correlation Heat-map between all numeric columns

An example of correlation is when two things are linked together in a mutually beneficial way. Correlation is useful in practically every context since it is more meaningful to explain something in terms of its connection to others. Data exploration is no exception to this as correlations help with seeing relationships between data columns; this can be done through numeric columns with the use of a heat-map.

2.3.6 Cluster size Analysis

The world is filled with an infinite amount of data and thus, it becomes very easy for all kinds of users to be slowed down by data overload. This never-ending influx of data necessitates a high-level view in order to adapt to the digital world. This can be achieved by grouping things together as groups of data allow us to look at the group’s first instead of the individual data points. Specific data science constructs can help with creating some groups out of a lot of data; this is otherwise called clustering or segmentation. Segmentation is the process of creating segments, which is a highly useful data exploration approach since it provides a clear picture of the data. To carry out segmentation, it is helpful to make an analysis of cluster size first so it can show you how data can be divided into various groups.

2.3.7 Clustering or Segmentation

After we determine the number of clusters we are going to use, we follow this first step by separating all the data into a specific number of clusters/segments which can further result in especially useful conclusions in data exploration.

2.3.8 Outlier overview

Outlier detection is defined as finding something atypical in the given data, this can also be defined as an anomaly detection. Outliers usually symbolize something unusual, rare or something extraordinary and they do not necessarily have a negative connotation. Outlier analysis helps us improve exploratory dataset research quality. This can be done through obtaining outlier values in numeric columns through standard deviation analysis or algorithms. Furthermore, an outlier overview analysis can give further information on the outliers in all the numeric columns.

2.3.9 Outlier analysis for multiple columns

Another important step of exploring the data is finding outliers based on multiple columns [17]. This can be achieved using different algorithms, one of them is the Isolation Forest.

2.3.10 Specialized Visualization

The most common visualization techniques are usually the classic ones, among which are the Bar chart, scatter plot etc. However, some other specialized visualizations are also of value during data exploration; these include Radar Chart, Neural Network visualization and Sankey charts. These visualizations help understand the data a lot better. Radar Chart encourages comparison between data whilst Neural Network visualization helps with understanding what combinations of columns can be

considered as important features. NNV can also help with understanding hidden/latent features. Finally, Sankey charts are useful in making path analysis.

2.4 Data processing and validation strategies for predictive models

Data mining algorithms: Neural Networks and SVM

There are 21 parameters in the multilayer perceptron technique used by this particular neural network classifier, which is called MLP Classifier. In order to do classification, this method (SVM) makes use of the Library for Support Vector Machines (libSVM). Additionally, the SVC contains a total of 14 variables.

Simulation steps: Model training, Prediction and evaluation

MLP Classifier and SVM employ the function fit (X, y) to fit the model as part of their model training, which is referred to as "model training." However, while discussing prediction, we employ the function predict(X) to assist us forecast the test data's target values. The test data and labels' mean accuracy (X) and labels' mean accuracy (y) are calculated using the score function "score(X,y)" (y). Lastly, a classification that is also utilized for evaluation may be expressed as a report (y true, y pred); this function generates a text report displaying the primary classification metrics (precision, fl-score, and recall).

Model validation: Classification metrics, cross-validation, K-fold

The term "classification metrics" refers to a collection of metrics computed using a confusion matrix and used to evaluate a model. Each class's accurate vs. wrong predictions are summarized in these data sets. Data mining techniques may be evaluated and validated via cross-validation. As a result, two datasets are created: one for training purposes and the other for testing. However, a K-fold cross-validation

approach is utilized to mitigate poor performance due to the random dataset partitioning of training and test data. The complete dataset (D) is split into k subgroups (D1, D2, Dk) of equal size, and each subset is randomized.

2.5 Data mining algorithms

Some power algorithms that will be useful for our predictions are listed below:

The randomly selected training sample is divided up using the **decision tree bagging technique** into several data subsets. Additionally, each subset is utilized to train decision trees, reducing the variance of a decision tree. Finally, this technique is able to deal with data that has a greater dimensionality.

Repeated boosting of decision trees results in a succession of decision trees. Decision trees are learned using instances that have been filtered out by prior trees. Every step of the way, it randomly fits consecutive trees and supports various loss functions. Over-fitting is a potential drawback of this technique.

In data mining, **the K-nearest neighbor** (KNN) algorithm is one of the most straightforward methods. Only if the test characteristics perfectly match one of the training samples will it be able to classify the data. KNN is very simple to use and may be beneficial in a wide range of scenarios. This approach, for example, is especially suitable for classes with several modes of communication. Because many test records do not perfectly match any training set, there is a lot of data loss when it comes to categorization.

One of the most used classification algorithms is the **Naive Bayes**. This algorithm develops rules that may be used to predict the class of future objects based on the classifications that have already been established. Additionally, it is simple to use and can be used to big datasets. NB's strengths include these aspects. Furthermore,

unlike the aforementioned approach of boosting decision trees, this algorithm does not need complicated and repetitious parameters.

In the family of supervised learning techniques, **support vector machines** (SVM) are used for classification, regression, and the identification of outliers. The purpose of SVM is to find the most accurate way to classify the training data into two groups. An advantage is that it takes a relatively modest number of training sets to get the desired results. SVMs, on the other hand, take a long time to train and test. They're also really complicated and need a lot of memory.

Unsupervised algorithms, on the other hand, employ iterative methods to divide a random dataset into multiple clusters (indicated by the letter "k"). Each observation or data point is linked to the closest cluster, calculated, and modified using these clusters. After then, the procedure is again, but with the updated changes, until the desired outcome is reached. It has the advantages of being simple to implement, quick, and low-cost computationally.

It is possible to create supervised artificial neural networks (ANNs) by the process of training them repeatedly on a test. They are often used in categorization and problem-prediction work. The input, intermediate, and output layers of an ANN are all distinct. They are very effective due to their ability to multitask seamlessly.

2.6 Methods for data interpolation and data extrapolation

2.6.1 Data interpolation

As the name suggests, interpolation is a technique for calculating unknown data values using known data. If you're interested in atmospheric research, interpolation methods like linear interpolation are the most widely employed.

The most often used data interpolation techniques are:

Inverse Distance Weighting: Basically, it's an enhanced version of the closest neighbour method. Date-specific values are calculated through a linear combination of nearby values.

Linear regression: Figure out how one or more explanatory factors are related to a predicted variable.

Probabilistic methods: Based on a correlation function, optimal interpolation relies on the output of numerical weather prediction models. This approach has the advantage of minimizing the predicted interpolation error, but the negative is that it is difficult to precisely quantify the error covariance. This algorithm determines the number of times your program is executed, which is dependent on the loop utilized in the program.

The Minimum Curvature approach is widely used in Earth science because it assumes that the interpolated surface formed by Minimum Curvature is equivalent to a linearly elastic plate that passes through all of the data values with the least degree of bending. The smoothest possible surface is achieved while striving to monitor your data as closely as possible with this approach.

Another method is called **The Modified Shephard's Method**, and it uses an inverse distance weighted least squares method. Unlike the Inverse Distance to a Power interpolator, this approach employs local least squares to remove or reduce the "target" look of the contours formed. With an accurate interpolator or with a smoothing interpolator, the modified Shephard's Method may be shown.

For interpolation**, kriging** is a geostatistical approach. There are three ways to use this approach to determine the accuracy of a forecast based on the estimated prediction error: Kruskal-Wallis in all its forms: simple, common, and universal.

Other methods- **MISH**: Uses interpolation to include data from time series. Interpolation is handled by MISH, and homogenization is handled by MASH.

PRISM: Point measurements of temperature, rainfall, and other climatic variables are used in this approach. To create maps and perform a variety of analyses, it is used with Geographic Information Systems (GIS).

2.6.2 Data extrapolation

Data extrapolation is limited to time-series forecasts and involves extrapolating previous patterns into the future to make statistical predictions. Extrapolation is helpful when unexpected big changes happen: when causal factors are expected to remain constant but they do not or when causal factors are not clearly understood in the context of a situation. An added benefit of this methodology is that it discourages the introduction of personal biases into the process. To put it another way, the way that extrapolation works is by interpolating a smooth nonlinear curve over all of the x values and then utilizing that curve to project future x values beyond the previous data set. A ratio of two polynomials is the outcome of either the polynomial functional form or the rational functional form.

The most used methods for data extrapolation are listed below:

Rule-Based Forecasting method: Rule Based Forecasting apply and develop rules using domain knowledge to get together different methods of extrapolation. It works based on domain knowledge.

Temporal extrapolation methods: Temporal extrapolation methods work based on prior understanding of the system and recent data to explain the developments for the future. It does not need many data requirements and it is easy to get estimates.

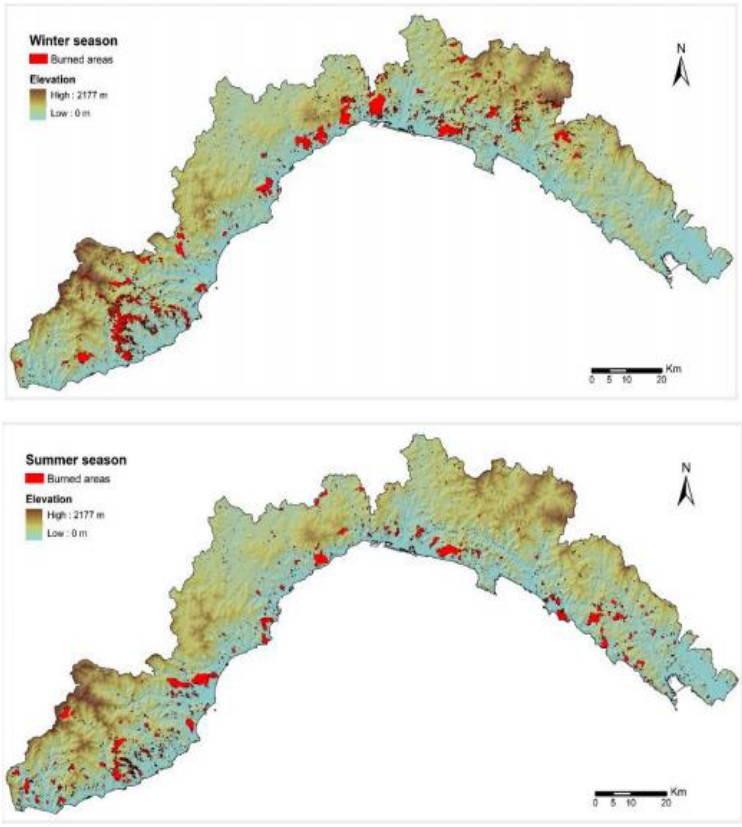
Auto-regression: Auto-regression is used for predicting error filtering and time series. It only works for linear relationships, has a great performance and replaces corrupted or missing samples of data.

CHAPTER III

WILDFIRE PREDICTIONS FOR AUSTRALIA

3.1 Datasets

Five datasets have been used for developing the predictive model for wildfire prediction in Australia during 2021. The datasets that have been incorporated are wildfires, historical weather forecast, historical weather, land classes and vegetation index.



The data is provided in CSV format as daily time series:

1. Historical wildfires
2. Historical weather
3. Historical vegetation index
4. Land classes (static throughout the contest)
5. Historical weather forecasts

Figure 3. Wildfire map during Winter and Summer Seasons

3.2 Analyzing the data

This part consists on analyzing the datasets that are used for building a predictive model for Australian wildfires from year 2005 to 2020. A description of datasets columns has been made, as well as the methods used for analyzing the data, explained in details earlier.

3.2.1 Historical Wildfires

The wildfire dataset contains fire cases in Australia starting from 2005. The data is processed as follows:

The data were spatially averaged to 7 regions or states in Australia. In addition to spatial aggregation, all data was aggregated by day starting from 1/1/2005. Multiple fires have been observed in each region at different timestamps during a single day. The numbers of flagged pixels for each day are reported in the count column. Furthermore, only fires that were identified by the algorithms with high confidence(>75%) are considered. More than 98% of all detected fires are presumed vegetation fires.

The dataset contains information about the seven regions, below are the listed columns:

Region: Seven regions of Australia (*Figure 4*)

Figure 4. Map of 7 regions in Australia

Mean_estimated_fire_brightness: Daily mean (by flagged fire pixels(=count)) of estimated fire brightness for presumed vegetation fires with a confidence level > 75% in Kelvin

Date: In UTC, provide the data for 24 hours ahead.

Estimated_fire_area: Daily sum of estimated fire area for presumed vegetation fires with a confidence > 75% for each region in km^2.

Mean_estimated_fire_radiative_power: Daily mean of estimated radiative power for presumed vegetation fires with a confidence level > 75% for a given region in megawatts.

Mean_confidence: Daily mean of confidence for presumed vegetation fires with a

confidence level > 75%.

Std_confidence: Standard deviation of estimated fire radiative power in megawatts.

Var_confidence: Variance of estimated fire radiative power in megawatts.

Count: Daily pixels for presumed vegetation fires have a confidence level of larger than 75% for a given region.

Replaced: Indicates with a Y whether the data has been replaced with standard quality data when they are available (usually with a 2-3 month lag). Replaced data has a slightly higher quality in terms of locations.

Starting to analyze our data: the shape of the data needs to be checked, as well as the data type in the dataset and finding the maximum and minimum. Further, null values will also be checked: if there are any, it has been tried to find the reason for those values. If there are no null values, the process is proceeded with cross-checking distinct Count Values when Std_confidence and Var_confidence are NULL.

	Region	Date	Estimated_fire_area	Mean_estimated_fire_brightness	Mean_estimated_fire_radiative_power	Mean_confidence	Std_confidence	Var_confidence
0	NSW	1/4/2005	8.68000	312.266667	42.400000	78.666667	2.886751	8.33333
1	NSW	1/5/2005	16.61125	322.475000	62.362500	85.500000	8.088793	65.42857
2	NSW	1/6/2005	5.52000	325.266667	38.400000	78.333333	3.214550	10.33333
3	NSW	1/7/2005	6.26400	313.870000	33.800000	92.200000	7.529640	56.70000
4	NSW	1/8/2005	5.40000	337.383333	122.533333	91.000000	7.937254	63.00000

Figure 5. Historical Wildfires data

```
df.loc[df.Var_confidence.isna(), :]
```

	Region	Date	Estimated_fire_area	Mean_estimated_fire_brightness	Mean_estimated_fire_radiative_power	Mean_confidence	Std_confidence
48	NSW	2005-02-26	1.00	303.15	8.0	79.0	NaN
149	NSW	2005-06-12	1.00	302.55	17.9	79.0	NaN
154	NSW	2005-06-18	5.27	301.30	71.9	77.0	NaN
157	NSW	2005-06-25	9.60	300.70	145.9	76.0	NaN
163	NSW	2005-07-09	2.80	294.65	37.8	79.0	NaN
...	...	...	...	...	...	...	...
26522	WA	2020-08-13	1.10	320.35	27.1	83.0	NaN
26523	WA	2020-08-14	1.00	302.15	15.8	77.0	NaN
26526	WA	2020-08-20	1.92	326.85	86.2	92.0	NaN

Figure 6. Cross-checking, null values

As it can be seen, **Standard Deviation** and **Variance Confidence** values are null because Count equals 1. A count of 1 shows that there was 1 pixel representing other values. So, if these NULL values are filled with zero, then unique regions can be viewed. So, the seven unique regions for Historical Wildfires are **NSW, NT, QL, SA, TA, VI,** and **WA.** Similarly, is imported the **Historical Weather** dataset to perform preprocessing to prepare the data for the next step. Similarly, as the steps performed on the first dataset, there is also the necessary to perform multiple steps on this dataset.

3.2.2 Historical Weather

The file HistoricalWeather.csv contains daily aggregates computed from the hourly ERAS climate reanalysis. If the data is in the format 'YYYY-mm-dd', the data in the row was created by aggregating the hourly ERAS data from "YYYY-mm-ddT00:00:00Z" to "YYYY-mm-(dd+l)T00:00:00Z". Region denotes an area in Australia. Raw ERAS data comes in raster form on a 0.25 x 0.25 degrees resolution grid. Following the temporal aggregation, the data were spatially aggregated. Some of the parameters are: Precipitation [mm/day], Relative humidity [%], Soil water content [m**3 m**-3], Solar radiation [MJ/day], Temperature [Cl, Wind speed (m/s)]. The raw ERAS data does not contain relative humidity. Instead, relative humidity data are computed from ERAS's temperature and dewpoint values.

Precipitation is derived from total precipitation. Hourly raw data is converted from m/hour to mm/hour. The relative humidity is derived from the temperature and dewpoint. Soil Water Content is given for 0 - 7 cm below the surface and wind Speed is calculated for every hour from the Easterly and Northerly 10-meter wind components.

It is better to rename columns to understand more clearly and then check for missing/null values in the dataset. While arranging the data and columns to look cleaner and more understandable, NULL values are cross-checked in the rearranged data. Furthermore, it is advised to perform some steps again for confirmation, such as checking NULL values to confirm that no value is missing while changing the dataset.

3.2.3 Historical Vegetation Index

Then Historical Vegetation Index dataset reports the monthly normalized differential vegetation index (NOVI) starting in 2005 for Australia. The dataset is based on the observations of the MODIS, Terra 13 satellite at 250 m resolution at 16 days intervals. Generally, the data can be assumed to be cloud free.

Region: The respective regions as mentioned above for which the data is aggregated.

Date: Month of acquisition of the data. All dates are in UTC and provide the data for the same months. If multiple timestamps are available during the month, the mean of the observation is computed first.

Vegetation_index_mean: The spatial mean of the vegetation index for the given region and month.

Vegetation_index_max: The maximum spatial value of the vegetation index for the given region and months.

Vegetation_index_min: The minimum spatial value of the vegetation index for the given region and months.

Vegetation_index_variance: The spatial variance of the vegetation index for the given region and months.

3.2.4 Land Class dataset

The next step of our analysis process would be working with the "Land Class" dataset. The "Land Class" dataset was derived from the CGLS land cover data product, based on PROBA-V satellite measurements.

With the source code below *(Figure 7*), A CSV file namely LandClass (**comma-separated values**) that has a specific format which allows data to be saved in a table structured format is accessed through the command “pd.read”. The results are displayed as shown in the figure below.

```
file_land = "Jan_09/LandClass.csv"
print("Reading file: '{}'".format(file_land))
df_land = pd.read_csv(file_land)
print("Loaded...")

df_land
```

```
Reading file: 'Jan_09/LandClass.csv'
Loaded...
```

	Region	Shrubs	Herbaceous vegetation	Cultivated and managed vegetation/agriculture (cropland)	Urban / built up	Bare / sparse vegetation	Permanent water bodies	Herbaceous wetland	Closed forest, evergreen, broad leaf	Closed forest, deciduous broad leaf	Closed forest, unknown	Open forest, evergreen broad leaf	Open forest, deciduous broad leaf	u del
0	NSW	6.2	43.6	13.0	0.3	0.2	0.2	0.1	14.7	6.8	0.3	0.5	3.7	
1	NT	18.1	48.9	0.1	0.0	0.4	0.1	0.1	1.0	7.7	0.1	0.1	13.6	
2	QL	9.5	45.3	1.6	0.1	1.1	0.1	0.0	5.3	13.3	0.3	0.1	12.0	
3	SA	24.1	54.8	5.8	0.1	4.8	1.2	0.1	0.3	1.3	0.1	0.1	1.4	
4	TA	0.7	23.8	1.2	0.2	0.1	1.9	1.4	50.1	0.6	1.1	7.2	1.5	
5	WA	31.3	43.5	5.6	0.0	1.0	0.4	0.0	1.2	2.4	0.2	0.1	4.8	
6	VI	1.4	35.0	23.3	1.0	0.1	0.6	0.3	23.9	3.8	0.3	1.5	2.7	

Figure 7. Land Class dataset columns

3.2.5 Historical Weather Forecasts dataset

"Historical Weather Forecasts" file contains daily aggregates calculated from NOAA's Global Forecast System (GFS) output.

The dataset contains the same parameters as Historical Weather's data. An extra column called "Lead time" shows the number of days the forecast is valid for.

Region: Represents a region in Australia. Raw ERAS data comes in raster form on a 0.5 x 0.5 degrees resolution grid. Following the temporal aggregation, the data were spatially aggregated.

Parameter: Precipitation [mm/day], Relative humidity[%], Solar radiation [MJ/day], Temperature [C], Wind speed [m/s].

Lead time [days]: Difference between the time the forecast is for ("valid time") and the time the forecast was made ("issue time" or "data time").

count()[unit: km"2]: Area ofthe Region. In km**2.

min(): Minimum value of the spatial aggregation.

max(): Maximum value of the spatial aggregation.

mean(): Average of the spatial aggregation. 9. variance(): 2nd moment of the spatial aggregation

The first step that is needed to perform would be: dropping "Area" values from Historical Weather Forecasts Data and also dropping duplicates from the file. As mentioned above in this paper the process of checking for duplicates

```
df_fires_weather_forcasts = wildfire_weather_data.merge(weather_forcasts_data, how='left', on=['Date', 'Region'])

# Number of records
num_rows, num_cols = df_fires_weather_forcasts.shape
print("Total Records:\t{}".format(num_rows))

df_fires_weather_forcasts.head()
```

```
Total Records:  44417
```

Figure 8. Merge of Wildfire Weather with Weather Forecasts data

The final arranged file is used for some insights to retrieve the needed information for the next process. Wildfires with Weather data has around 26K data, whereas when combined with Weather Forecasts data, it results in ~44K records because Weather Forecasts data contains Lead Time values of 5, 10, and 15 on the basis of some dates and regions. Historical Weather Forecasts data start in 01-01-2014, and thus, it needs to filter the dataset accordingly.

Pandas DataFrame loc[ ] is allowing us to access a group of rows and columns (*Figure 9*). We can pass labels as well as boolean values to select the rows and columns.

```
df_analysis.loc[df_analysis['Lead time'].isna(), :]
```

	Region	Date	Estimated_fire_area	Mean_estimated_fire_brightness	Mean_estimated_fire_radiative_power	Mean_confidence	Std_confidence
37	NSW	2014-02-11	11.128571	328.592857	57.742857	88.142857	8.474050
83	NSW	2014-04-15	133.031250	310.690625	128.181250	85.406250	6.964701
84	NSW	2014-04-16	49.102083	314.197917	36.979167	85.583333	5.978092

Figure 9. Checking for the reason of null values

3.3 Data Visualization

The ability to discover and convey real-time patterns, outliers, and fresh insights about the data is made possible through data visualization. Data visualization in python is perhaps one of the most utilized features for data science with python in today's day and age. The libraries in python come with lots of different features that enable users to make highly customized, elegant, and interactive plots.

Visualization Libraries are imported to perform the visualization of the data. Different libraries can be used for visualizations; in this case, "Seaborn" has been utilized. Seaborn is a dataset-oriented library for making statistical representations. It is developed atop matplotlib and to create different visualizations. It is integrated with pandas data structures. The library internally performs the required mapping and aggregation to create informative visuals. By using Seaborn, different plots for exploratory data analysis are plotted and by looking at them, decisions can be made and get to know the insights better (*Figure 10*).

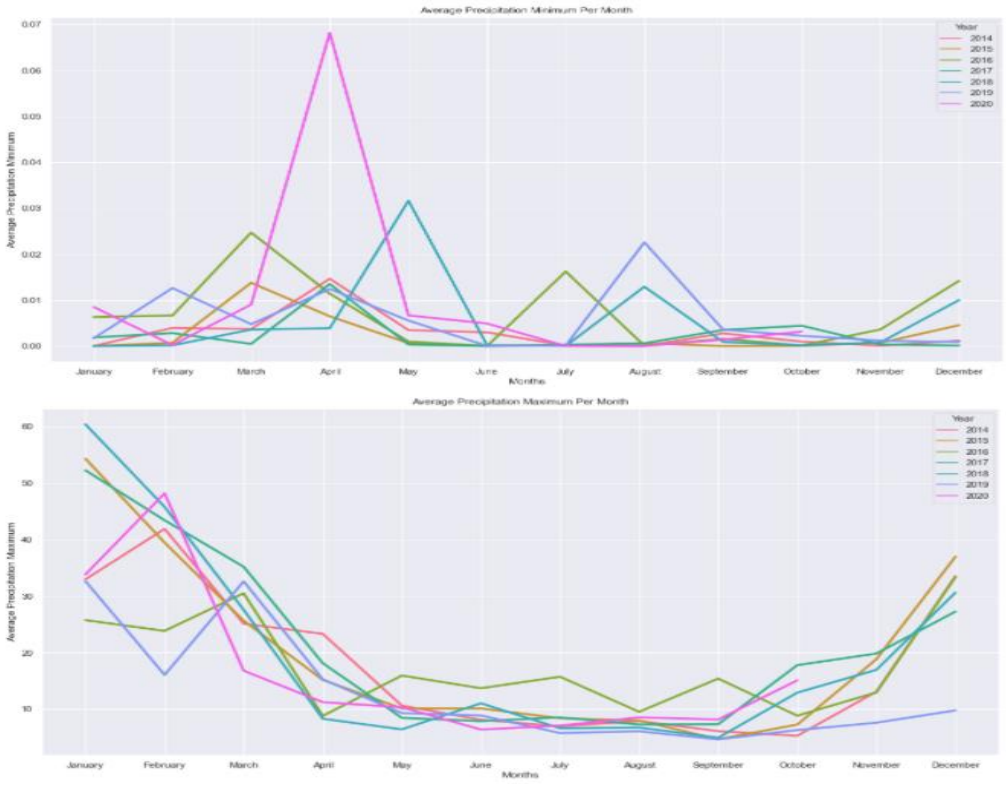


Figure 10. Average Precipitation Min and Max per Month

3.4 Data Preprocessing

3.4.1 Weather Dataframe

The dataset contains daily weather statistics for the seven regions of Australia. The type of weather includes Precipitation, Soil Water Content, Relative Humidity, Solar Radiation, Temperature, Wind speed, and the type of statistics include min, max, mean, and variance.

However, for this work, it would be assumed the mean weather parameter (for example, temperature) will be uniform across the region. It has been chosen to do so due to the complexity of handling different means at the district level (2nd administrative level).

Therefore, the ONLY mean statistics will be taken for the weather parameters and transformed into a data frame that will merge well with the other datasets (vegetation, wildfires).

3.4.2 Wildfires Dataframe

The wildfires dataframe is based on the MCD14DL dataset, and the "estimated_fire_area" is calculated by multiplying the scan and track values from the MCD14DL. Scan and track values were required due to the increasing pixel resolution as the pixel reaches closer to the picture's end. The "estimtaed_fire_area" is the Y dependent for the forecast for 2021 February. In addition, we will also take the 'count' of the fires to do some feature engineering later. The count represents the number of pixels originally found on the MOD14AL1/MYD14AL1 satellite images.

3.4.3 Vegetation Index

The vegetation dataset contains NDVI statistics separated by region. The statistics include mean, max, min, standard deviation, and variance. Similar to the

weather dataset, it would be assumed that the mean NDVI is uniform across each region. Although the NDVI values might differ drastically across the region, this assumption has been proposed here to simplify the problem. The vegetation index is in a monthly format, and it would need to interpolate to a daily format. For days after 12/1/2020, the simple FBProphet model will be used to forecast up to 01/22 (January 22) as the last day of recording wildfires. It will need iteration over each region to fit an FBProphet model for each region. Here we create a function first to slice the dataset over the Region column, setting the data frame to index by DateTime.

The code below (*Figure 11*) returns a dataframe with the specified regions. The second part is a Callable that returns a boolean Series and slice with labels for row and single label for column. .loc is label-based, which specify rows and columns based on their row and column labels. The datetime() class (constructor) of the datetime module is used to determine date. The pd.pivot function returns reshaped DataFrame organized by given index / column values. It uses unique values from specified *index* / *columns* to form axes of the resulting DataFrame. This function does not support data aggregation, multiple values will result in a MultiIndex in the columns.

```
def region_dataframe(region):
    """
    returns dataframe with the following related to specific region:
    estimated_fire_area
    mean_Precipitation
    mean_RelativeHumidity
    mean_SoilWaterContent
    mean_SolarRadiation
    mean_Temperature
    mean_Windspeed
    vegetation_index_mean
    """

    # ^^^^^ START of WILDFIRES ^^^^^

    # Slicing the wildfires dataset to contain on the specified region
    wildfires = pd.read_csv('Jan_09/Historical_Wildfires.csv')
    wildfires = wildfires.loc[wildfires['Region'].eq(region)].copy()

    # Pivot the dataframe to index by datetime
    wildfires['Date'] = pd.to_datetime(wildfires['Date'])
    wildfires = pd.pivot(wildfires, index = 'Date', columns = 'Region', values = ['Estimated_fire_area', 'Count'])
    columns = []
    for i,x in wildfires.columns:
        columns.append('{}_{}'.format(i,x))
    wildfires.columns = columns
```

Figure 11. Slice the dataset over the Region column

After following multiple steps analyzing and preprocessing data we will achieve a model that forecasts data for the next 52 days. As a final result all the datasets with axis = 1 (column-wise) and their returning dataframe are concatenated.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import datetime
import os

from sklearn.preprocessing import MinMaxScaler, RobustScaler, StandardScaler
import tensorflow as tf
from tqdm import tqdm
```

Figure 12. Necessary imports

(*Figure 12*) The first thing to do is to import the Python libraries. *"Import pandas as pd"* presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a NumPy matrix array.

The "import numpy" portion of the code tells Python to bring the NumPy library into the current environment. The "as np" portion of the code then tells Python to give NumPy the alias of np. This allows the use of NumPy functions by simply typing np.function_name rather than numpy.function_name.

"import matplotlib.pyplot as plt" gives an unfamiliar reader a hint that pyplot is a module, rather than a function which could be incorrectly assumed from the first form.

Seaborn is one of the libraries we need to import as well. By convention, it is imported with the shorthand sns.

Behind the scenes, seaborn uses matplotlib to draw its plots. For interactive work, it's recommended to use a Jupyter/IPython interface in matplotlib mode, or else you'll have to call matplotlib.pyplot.show() when you want to see the plot. *%matplotlib inline* sets the backend of matplotlib to the 'inline' backend: With this backend, the output of plotting commands is displayed inline within frontends like the Jupyter notebook, directly below the code cell that produced it. The resulting plots will then also be stored in the notebook document.

Date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time through the command *import datetime*.

The OS module provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc. The line "*import os"* import the "os" module to interact with the underlying operating system.

The graph below (*Figure 13*) visualize the estimated fire area based on seven different regions of Australia. As it is shown, each region has different spread rate throughout the years. We can notice that the regions of Victoria and New South Wales have the highest values for the year 2021.

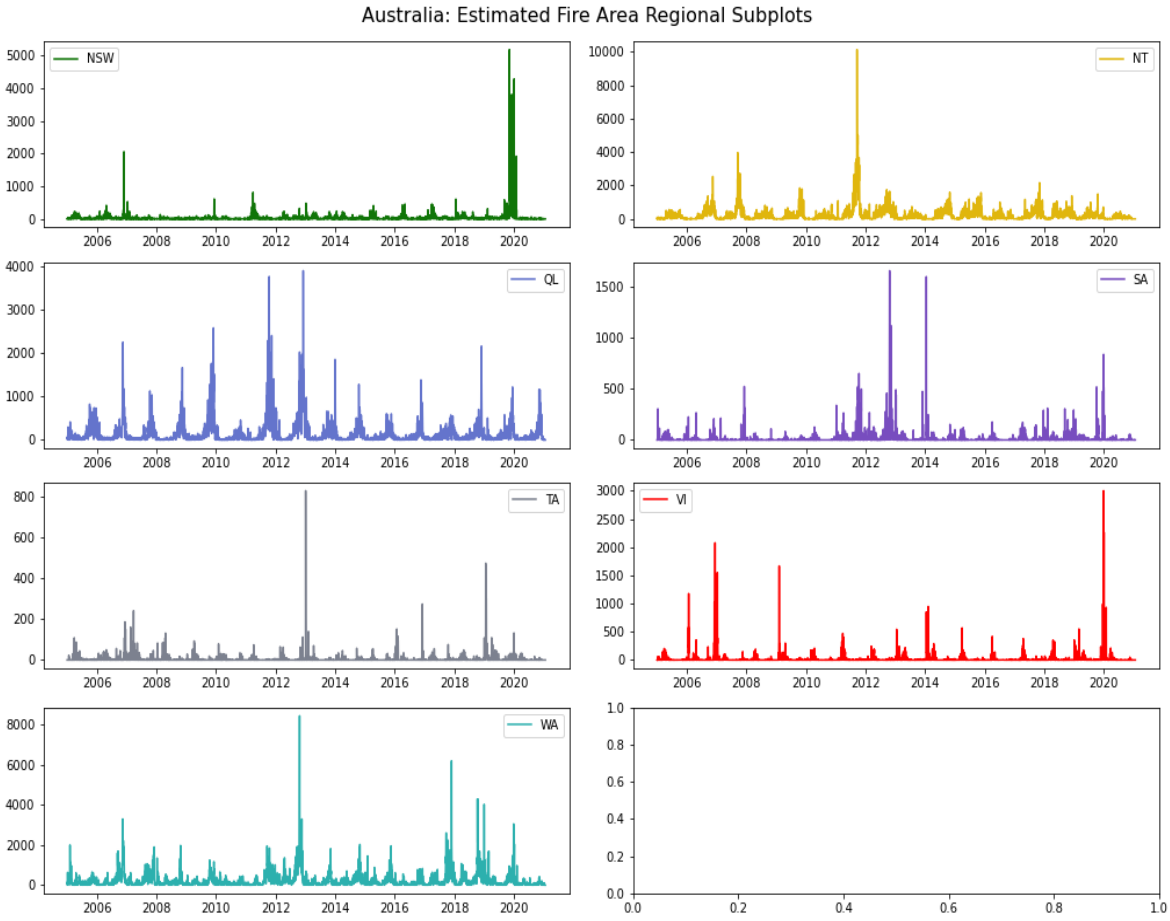


Figure 13. Estimated Fire Area Regional Subplots

3.5 Feature Engineering

Feature engineering is the process of extracting characteristics from raw data using domain knowledge. A feature is a characteristic shared by independent units on which evaluation/prediction is performed in this procedure. Predictive models frequently use these characteristics, and they have an impact on the outcomes and implications. Feature engineering has two aims: 1) Produce the proper input dataset, which must be compatible with the machine learning algorithm requirements, and 2) Boosting the performance of machine learning models.

We transform the estimated fire area to log scale and exponentiation after predicting in log scale form. Surface Area: Assume the fire areas are conglomerated into one pixel, and the surface area would be 4 * square root of the Area. Assume the fire area pixels are separated (non-touching), and the areas of each pixel are the same. The surface area would be the count of the pixels * square root of (area/count).

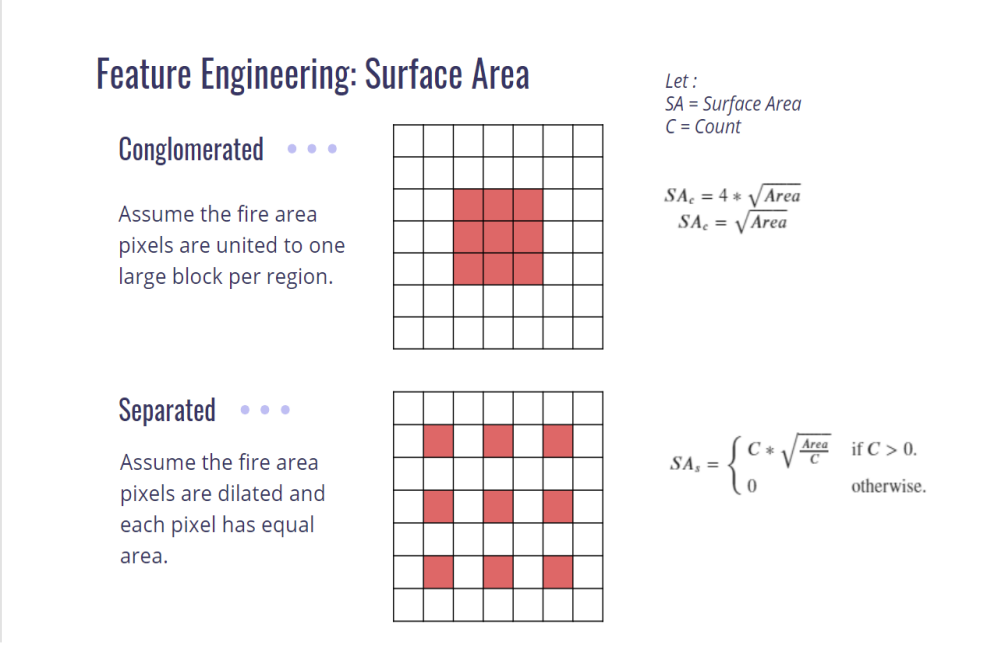


Figure 14. Future Engineering: Surface Area

3.6 Test and train data

Testing data should be unlabeled as training and validation data include labels that monitor the performance metrics of the model. This kind of test data is useful as it provides a final, real-life check of an unknown dataset so that it can confirm that the ML algorithm was trained sufficiently and effectively.

One of other important processes would be normalizing data using min-max scaler(). MinMax Scaler transform features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set. In the process of separating the train test split, the values of the test set will be validated from the input of the last 120 days of the train set.

MinMaxScaler has been chosen to retain the original distribution shape while scaling the features to a range value of 0 and 1.

In the image below train_df.dscribe(include = [:' ']) pulls out the objects dtypes attributes and shows their count/frequency/max/quartiles.

“train_test_split” is used for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.

The merge() method updates the content of two DataFrame by merging them together, using the specified method(s).

Train test split

```
train_df = merge_df[:'2020-10-28']
# 120 days before 2019-12-01
val_df = merge_df['2020-07-01':'2020-12-08']
# 161 days(120 + 41 days) before 2021-01-18
test_df = merge_df['2020-08-11':]
```

MinMax Scaler

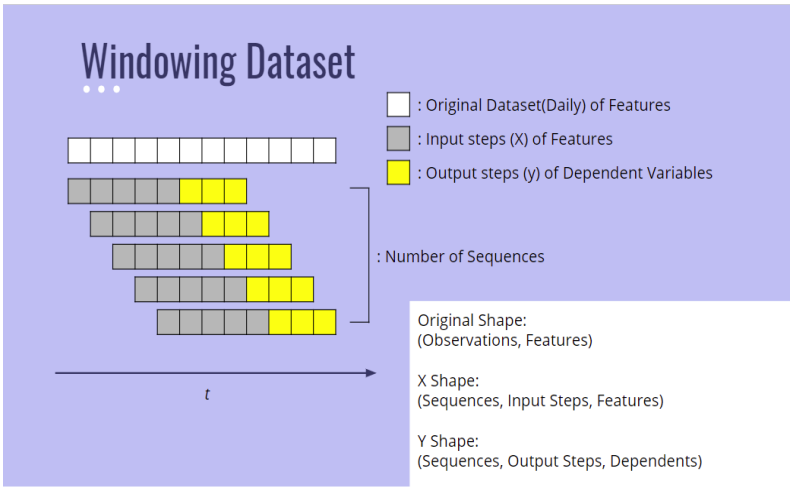
```
scaler = MinMaxScaler()
# scaler = StandardScaler()
scaler.fit(train_df)
scaled_train_df = scaler.transform(train_df)
train_df = pd.DataFrame(data = scaled_train_df,
                        index = train_df.index,
                        columns = train_df.columns)

scaled_val_df = scaler.transform(val_df)
val_df = pd.DataFrame(data = scaled_val_df,
                        index = val_df.index,
                        columns = val_df.columns)

scaled_test_df = scaler.transform(test_df)
test_df = pd.DataFrame(data = scaled_test_df,
                        index = test_df.index,
                        columns = test_df.columns)
```

Figure 15. Train-test split, Minmax Scaler

Because slicing, which is a feature that enables accessing parts of sequences like strings, tuples, and lists, does not store static shape information, one must manually set the forms. This will allow the datasets to be inspected more easily.



An important step before implementing DCNN model , which will be explained later , is creating the properties for the window of dilated CNN for the test, validation, and train.

Figure 16. Windowing dataset

3.7 DCNN Model

The Deep Convolutional Neural Network (DCNN) is a specific model that was recently applied to classify large datasets. This model is exceptionally well-programmed as it is able to learn simple filters on its own and further hierarchically combine them. An interesting benefit of DCNN is their layering: the model uses a three-dimensional neural network that receives data as input and further uses it to train a classifier. This network has four different layers: convolution, activation, and fully connected, discussed below.

3.8 Convolutional Layer

It uses a convolution filter to identify patterns. This is how it all works: A convolution— The neural network's inputs are multiplied by a set of weights during the convolution process. Kernels or filters— A kernel or filter iterates through data many times as part of the multiplication process. Dot or scalar product— A mathematical operation is carried out during convolution. The filter is used to multiply the weights with various values. Filter locations are determined by summing all of the input values.

3.9 ReLU Activation Layer

A nonlinear activation layer, such as Rectified Linear Unit, is used to process the convolution maps. In this layer we remove every negative value from the filtered image and replace it with zero. This function only activates when the node input is above a certain quantity. So, when the input is below zero the output is zero. However, when the input rises above a certain threshold it has *linear*

relationship with the dependent variable. This means that it is able to *accelerate the speed of* a training data set in a deep neural network that is faster than other activation functions – this is done to avoid summing up with zero.[]

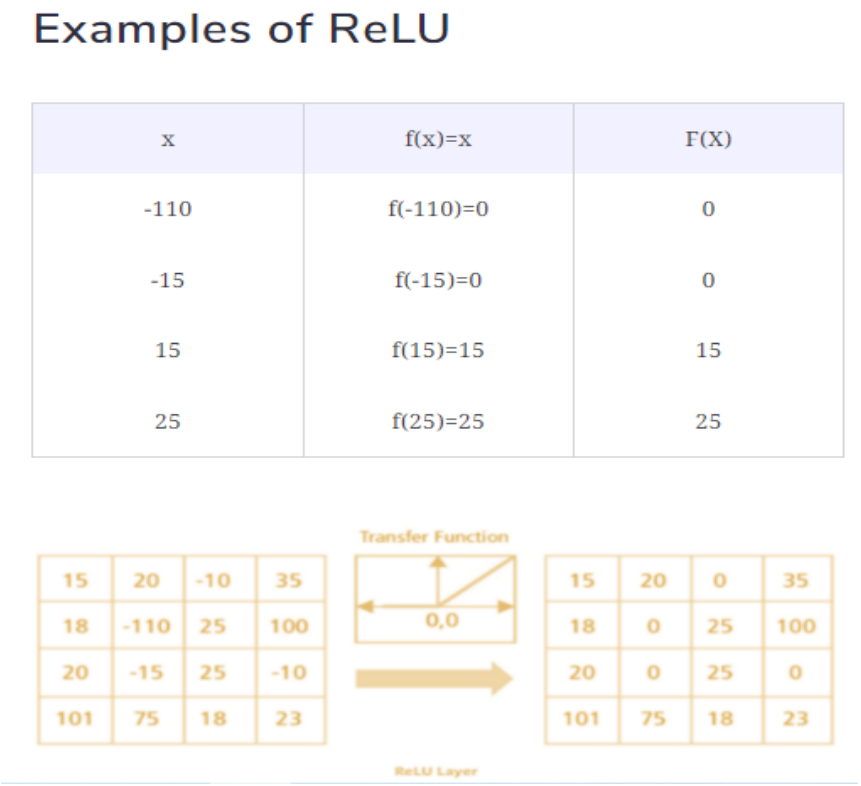


Figure 17. Examples of ReLu

3.10 Pooling Layer

Pooling layers are significant for this model as they reduce the size of the data and maintain/extract only the most important information. Reducing the number of calculations and parameters in the network also helps reduce over-fitting. Convolution and pooling layers are iterated many times until we get at a multi-layer perceptron of "completely linked" neural networks at the network's ultimate end.

3.11 Fully Connected Layer

Activation and pooling layers are interspersed across DCNN frameworks, allowing for several fully linked layers to exist. A softmax function is applied at the end of the outputs of the ultimately linked layers.

During DCNN we fit the model once, with an increased learning rate from 1e-6 to 1e-1 to find a stable learning rate for the final model. After visually locating the learning rate, it has been decided to refit the model over 1500 epochs.

A learning rate between 1e-5 and 1e-4 seemed appropriate as the loss began fluctuating between 1e-4 and 1e-3.

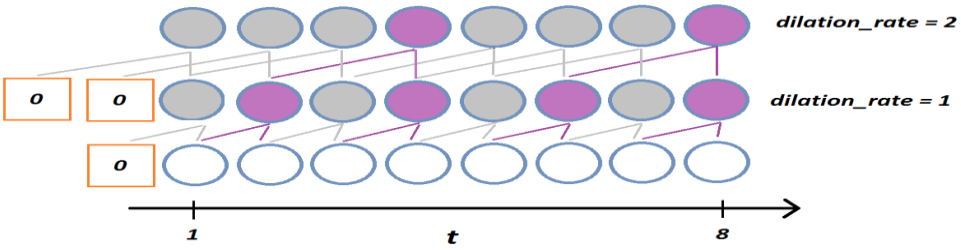


Figure 18. DCNN example

CHAPTER IV

FINAL PREDICTION FOR DECEMBER 2021

The final phase would be after the data is compiled and fits to the model data. The model is known as the baseline model. Baseline model should be simple. Simple models are less likely to overfit, as the complexity may kill the performance. The model should be interpretable. That's the purpose of using baseline model in this work.

By subclassing the Model class: in that case, layers have been defined and you should implement the model's forward pass in call function. The class is a user-defined blueprint or prototype from which objects are created.

We have used timeseries data with variables, trying to predict them as illustrated in the code below.

Baseline(Last)

```
class MultiStepLastBaseline(tf.keras.Model):
    def call(self, inputs):
        return tf.tile(inputs[:, -1:, :], [1, n_steps_out, 1])

last_baseline = MultiStepLastBaseline()
last_baseline.compile(loss = tf.keras.losses.Poisson(),
                      metrics = ['mae'])

multi_window.plot(last_baseline)
```

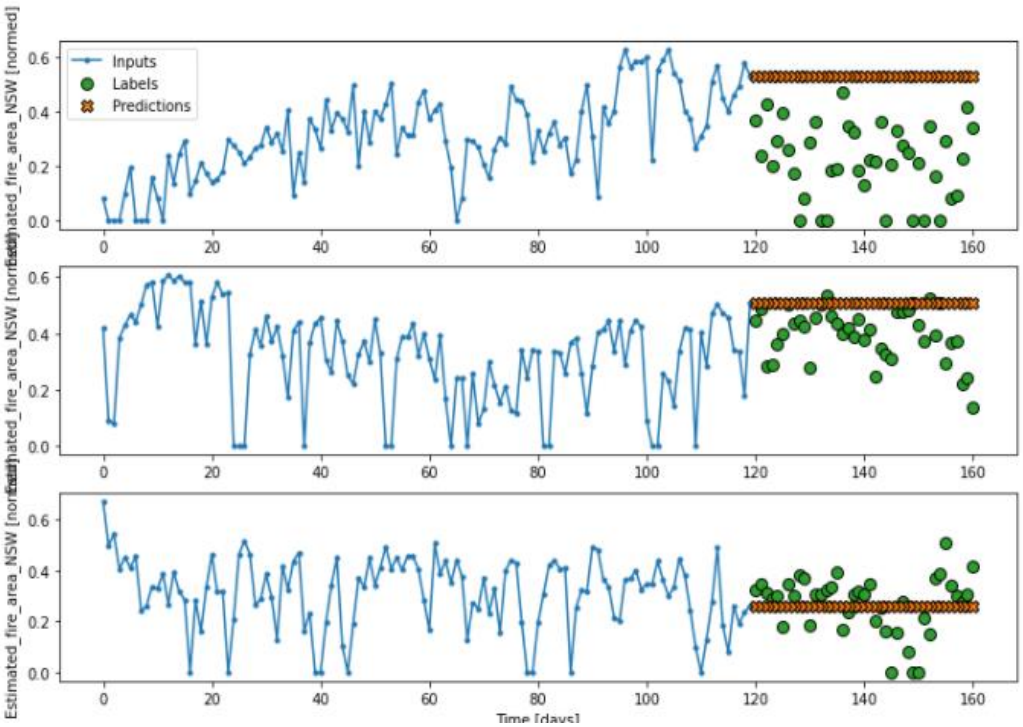


Figure 19. Baseline model

Initially, the preprocessing equivalent to time-distributed dense is executed. Second, filtering convolution to then multiply the filter and get the branches is performed. Lastly, post-preprocessing equivalent to time-distributed dense is executed.

A 1D dilated causal convolutional neural network was deployed as the model. The work resulted to good predictions, and the link was found to a GitHub repo for a similar work by Joseph Eddy [33]. He used a Seq2Seq LSTM and WaveNet-inspired algorithm to analyze web traffic statistics.

```
n_features = 77
n_output = 77 * 41
n_filters = 32
filter_width = 2
dilation_rates = [2**i for i in range(8)]

tf.keras.backend.clear_session()
history_seq = tf.keras.layers.Input(shape =(n_steps_in, n_features))
x = history_seq

skips = []
for dilation_rate in dilation_rates:

    # preprocessing - equivalent to time-distributed dense
    x = tf.keras.layers.Conv1D(16, 1, activation='relu', kernel_initializer = tf.keras.initializers.GlorotNormal(seed=42))(

    # filter convolution
    x_f = tf.keras.layers.Conv1D(filters=n_filters,
                                 kernel_size=filter_width,
                                activation = 'tanh',
                                 padding='causal',
                                 dilation_rate=dilation_rate)(x)

    # gating convolution
    x_g = tf.keras.layers.Conv1D(filters=n_filters,
                                  kernel_size=filter_width,
                                  activation = 'sigmoid',
                                  padding='causal',
                                  dilation_rate=dilation_rate)(x)

    # multiply filter and gating branches
    z = tf.keras.layers.Multiply()([x_f, x_g])

    # postprocessing - equivalent to time-distributed dense
    z = tf.keras.layers.Conv1D(16, 1, activation='relu')(z)


    # residual connection
    x = tf.keras.layers.Add()([x, z])


    # collect skip connections
    skips.append(z)
```

Figure 20. DCNN Slingshot

In this phase, all skip connection outputs are added. It shows the final time distributed dense layers. Using Matplotlib, plots between actual values and predicted values are created after the algorithm's training.

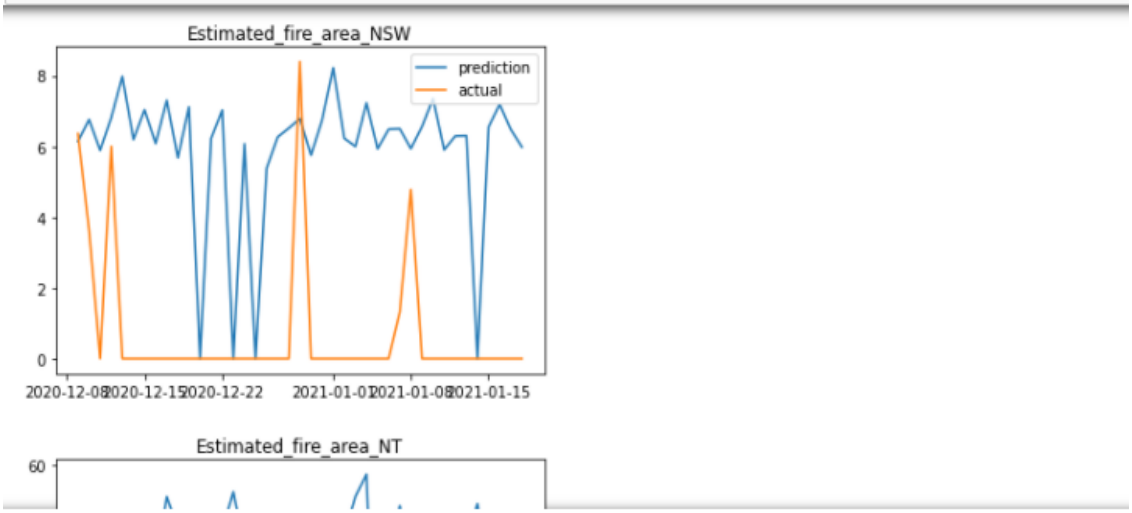
The plot below (*Figure 21*) reflects the predicted values and actual values for Australian wildfires during different months (date being the dependent variable).

The “inverse transform” method is used for scaling back the data to the original representation, while “numpy.exp” calculates the exponential of all elements in the array.

```
conv_single_df = pd.DataFrame(conv_single_pred_region,
            index = test_df[-41:].index,
            columns = fire_regions)
conv_sling_df = pd.DataFrame(conv_sling_pred_region,
                             index = test_df[-41:].index,
                             columns = fire_regions)
```

```
match_test = pd.DataFrame(np.exp(scaler.inverse_transform(test_df[-41:])[-41:, :7]) - 1,
                          index = test_df[-41:].index,
                          columns = fire_regions)
```

```
for i in fire_regions:
    plt.plot(conv_sling_df[i], label = 'prediction')
    plt.plot(match_test[i], label = 'actual')
    plt.legend()
    plt.title(i)
    plt.show();
```




```
for i in fire_regions:
    plt.plot(conv_single_df[i], label = 'prediction')
    plt.plot(match_test[i], label = 'actual')
    plt.legend()
    plt.title(i)
    plt.show();
```

Figure 21. Estimated fire area - prediction

In this phase, the estimated fire region ranging from 2020-12-09 to 2021-12-08 is plotted and displayed.

```
conv_single_df = pd.DataFrame(conv_single_pred_region,
            index = pd.date_range(start = '2020-12-09', end = '2021-12-08'),
            columns = fire_regions)
```

```
for i in fire_regions:
    plt.plot(conv_single_df[i])
    plt.show();
```

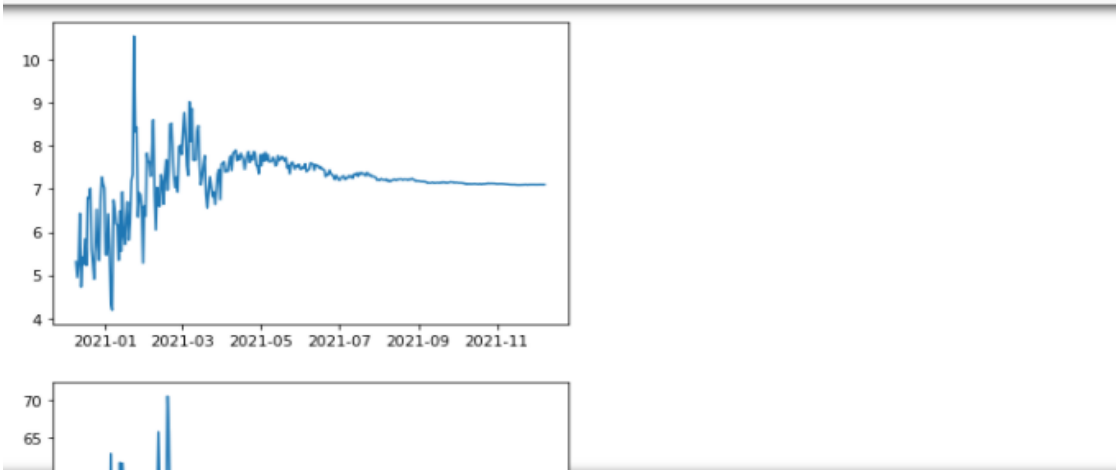


Figure 22. Fire regions in December 2021

The accuracy of the model was examined using classification metrics, cross-validation, and regularization, and the model produced good results for the algorithms utilized. This model's performance was further demonstrated by comparing other wildfire prediction models. All of these findings support the model's ability to predict the occurrence of wildfires.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

5.1. Conclusions

Taking into consideration the research done in this paper, alongside the literature review, the conclusion seems to be an obvious one; that is, that big data, machine learning ultimately can help with reducing the consequences of natural disasters. Further, the above-mentioned developed techniques can assist in six different areas of disaster management, including early warning damage detection and assessment, monitoring and detection of disaster effects, post-disaster coordination and response planning, and long-term risk assessment or even complete risk mitigation.

This project has dealt with a local problem and a worldwide one as the tide of climate change threatens to bring destruction to our lives and ecosystems. Every year, we seem to be confronted with the fact that some forest, somewhere in the world, is being destroyed by fire. Burning thousands of hectares each year equals the amount of forest lost to logging and agriculture combined, according to figures. We must realize that forest fires not only destroy the structure and composition of forests, but they also open them up to invading species, endanger biological variety, change water cycles or soil fertility, or even cause the extinction of humans who live there. Therefore, in order to decrease the damage done by these kinds of disasters, we can implement a solution, as described in this project, which integrates the prediction of wildfires through using newly developed data science techniques such as big data.

Throughout this project, we have shown how this solution is possible as we identified the studied area, collected the corresponding data, preprocessed them and finally saved the necessary datasets, these were further analyzed using data mining algorithms. After successfully running the simulation, the model seemed to predict

relatively realistic data for December 2021. Therefore, after putting it to the test, we can also conclude that this is a valid way for predicting future natural disasters and especially fires, with a close model accuracy. This mean that this algorithm or a similar one can be used to predict fires in other areas or can be adapted to forecast other natural disasters.

5.2. Future Work

In this research, a pressing issue that affects both our personal well-being and the health of our natural environment was addressed. By using the data and developing predictive models about wildfire behavior, we hope to lessen the impact of this calamity and save lives and property. A proper extraction and usage of this data is required.

A way to construct datasets has been developed. After determining the study region, we used two of the most well supervised data mining methods to examine it: Networks of SVMs and neural networks. The model's accuracy was evaluated using classification metrics, regularization, and cross-validation. The model's ability to anticipate wildfires was also shown by comparing it to other models. As can be seen from these findings, the model is quite good at forecasting the occurrence of wildfires. By including meteorological data into the model, we hope to improve it in the future. Wildfires are caused by a variety of factors, including the weather. As a result, its elimination becomes much more difficult because of its influence on the power and movement of fire. Wildfires may be affected by three weather variables: Temperature, Wind, and Water Content of the Soil.

The work for the future would also consist of strengthening the model by including more data in relation to fires and other disasters and further improving its accuracy through replications of this project or other critiques from people in the field.

REFERENCES

[1] B. W. Robertson, M. Johnson, D. Murthy, W. Roth Smith, K. K. Stephens, "Using a combination of human insights and 'deep learning' for real-time disaster communication", Published by Elsevier Ltd on (http://creativecommons.org/licenses/bync-nd/4.0/, 2019.

[2] S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti, "A Review on Application of Data Mining Techniques to Combat Natural Disasters", https://doi.org/10.1016/j.asej.2016.01.012, 2018.

[3] "BIG DATA FOR CLIMATE CHANGE AND DISASTER RESILIENCE: REALISING THE BENEFITS FOR DEVELOPING COUNTRIES" report, 2015.

[4] Y. Sayada, H. Mousannifb, H. Al Moatassimea, "Predictive modeling of wildfires: A new dataset and machine learning approach", published https://doi.org/10.1016/j.firesaf.2019.01.006, 2019.

[5] P. S. M. Nozari, "The Potential of Data Analytics in Disaster Management", DOI: 10.1007/978-3-030-03317-0_28, 2019.

[6] R. Arinta, Andi W.R. Emanuel, "Natural Disaster Application on Big Data and Machine Learning: A Review", 2019.

[7] A. A. Alkhatib, "A Review on Forest Fire Detection Techniques", International Journal of Distributed Sensor Networks Volume 2014, Article ID 597368, 2019.

[8] M. Tonini , M. D'Andrea , G. Biondi , S. Esposti , A. Trucchia and P. Fiorucci, "A Machine Learning-Based Approach for Wildfire Susceptibility Mapping". The Case Study of the Liguria Region in Italy, Geosciences 2020, 10, 105; doi:10.3390/geosciences10030105, 2020.

[9] P. Jain, S. C.P. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan, "A review of machine learning applications in wildfire science and management", Canada, Published at www.nrcresearchpress.com/er , 28 July 2020.

[10] Y. Li, Z. Feng, Ziyu Zhao, S. Chen, H. Zhang, "Research on Multi-factor Forest Fire Prediction Model Using Machine Learning Method", China, 2019.

[11] S. Sala, "Using Big Data to detect and predict natural hazards better and faster: lessons learned with hurricanes, earthquakes, floods", 2016.

[12] S. Cee, "Using Big Data To Predict Natural Disasters", retrieved from http://deeptechwire.com/using-big-data-to-predict-natural-disasters/ , 2019.

[13] Chin-Shung Yang, Szu-Pyng Kao, Fen-Bin Lee, Pen-Shan Hung, "TWELVE DIFFERENT INTERPOLATION METHODS: A CASE STUDY OF SURFER 8.0", National Chung Hsing University, 2019.

[14] R. Prakash, S. Mathangi, U. Acharya, Z. Noorain, K. Horadi, "Prediction, Analysis, And Relief Measure Reports For Disaster Crisis Management Using Regression, Artificial Neural Network, And RFC", European Journal of Molecular & Clinical Medicine Volume 07, 2020.

[15] Carol K. Joseph, S. Kakade, "Predicting Impact of Natural Calamities in Era of Big Data and Data Science", 7TH INTERNATIONAL CONGRESS ON ENVIRONMENTAL MODELLING AND SOFTWARE, 2019.

[16] Manzhu Yu, Chaowei Yang , Yun Li I, "Big Data in Natural Disaster Management", Review, 2018.

[17] S. Kaler, A. Sharma, "Big Data Predicting Natural Disasters", Journal of Emerging Technologies and Innovative Research, 2018.

[18] M. Arslan, A. Roxin, C. Cruz, Dominique Ginhac, "A Review on Applications of Big Data for Disaster Management", The 13th International Conference on SIGNAL IMAGE TECHNOLOGY & INTERNET BASED SYSTEMS, 2017.

[19] Amir Elichai, "How Big Data Can Help in Disaster Response", 2018.

[20] Joice K. Joseph, Karunakaran Akhil Dev , A.P. Pradeepkumar, Mahesh Mohan, "Big Data Analytics and Social Media in Disaster Management", 2018.

[21] S. Cee, "Using Big Data To Predict Natural Disasters", 2019.

[22] G. Anadiotis, "Raiders of the storm: The data science behind weather prediction", 2017.

[23] M. Lavrskyi, "Data Science Usage in Natural Disasters Predictions", 2019.

[24] Zh. L. ,Takano K. , Ji Y. ,Yamada S., "Big Data Analytics for Disaster Preparedness and Response of Mobile Communication Infrastructure during Natural Hazards", American Geophysical Union, Fall Meeting, 2015.

[25] D. Matthews, "How Big Data and IoT are Helping Natural Disaster Predictions (and Relief)", 2018.

[26] P. Gardoni, C. Murphy, "Using opportunities in big data analytics to more accurately predict societal consequences of natural disasters", 2019.

[27] N. Hurst, "How Satellites and Big Data Are Predicting the Behavior of Hurricanes and Other Natural Disasters", 2018.

[28] S. Goswamia, S. Chakrabortya, S. Ghosha, A. Chakrabartib, B. Chakraborty, "A review on application of data mining techniquesto combat natural disasters", 2016.

[29] JOSIE GARTHWAITE, "AI detects hidden earthquakes", 2020.

[30] Glantz, M., & Mun, J., Projections and Risk Assessment. Credit Engineering for Bankers, 185–236. doi:10.1016/b978-0-12-378585-5.10008-9, 2011.

[31] R. Rizki Arinta, A. W.R. Emanuel, "Natural Disaster Application on Big Data and Machine Learning: A Review", 4th International Conference on Information Technology, Information Systems and Electrical Engineering, 2019.

[32] N.A. Sundar, P.P. Latha, M.R. Chandra, Performance analysis of classification data mining techniques over heart disease data base, Int. J. Eng. Sci. Adv. Technol. 2 (3) (2012) 470–478

[33] Joseph Eddy. “High Dimensional Time Series Forecasting with Convolutional Neural Networks”. Available: https://github.com/JEddy92/TimeSeries_Seq2Seq/blob/master/notebooks/TS_Seq2Seq_Conv_Intro.ipynb.

[34] [Online]. Avaliable: https://developer.ibm.com/blogs/call-for-code-spot-challenge-for-wildfires-predictions-comparing-approaches/#:~:text=The%20Call%20for%20Code%20Spot%20Challenge%20for%20Wildfires,-The%20wildfire%20prediction&text=The%20goal%20of%20the%20challenge,data%2C%20updated%20to%20January%2029.

[35] P. Barmpoutis, P. Papaioannou, K. Dimitropoulos and N. Grammalidis "A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing", Review, Greece, 2020.

[36] E. Chuvieco, F. Mouillot, G. R.van der Werf, J. San Miguel, M. Tanase, N. Koutsias, M. García, " Historical background and current developments for mapping burned area from satellite Earth observation", *Remote Sensing of Environment*, vol. 255, 10.1016, 2019.

[37] C. Gómez, J. C.White, M. A.Wulderc, "Optical remotely sensed time series data for land cover classification: A review", vol. 116, 10.1016, 2016.