



Luís Duque Santos

Master of Science

GPU Accelerated Classifier Benchmarking for Wildfire Related Tasks

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Informatics Engineering

Adviser: Carlos Viegas Damásio, Associate Professor,
NOVA University of Lisbon

Co-advisers: Pedro Medeiros, Associate Professor,
NOVA University of Lisbon
Susana Nascimento, Assistant Professor,
NOVA University of Lisbon

Examination Committee



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

August, 2017

GPU Accelerated Classifier Benchmarking for Wildfire Related Tasks

Copyright © Luís Duque Santos, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

ACKNOWLEDGEMENTS

In First place, I would like to thank my adviser Professor Carlos Viegas Damásio for his help and guidance during the elaboration of this thesis, as well as to my co-advisers, Professors Pedro Medeiros and Susana Nascimento for their insights relating to their areas of expertise. I extend my thanks to Professor João Moura Pires and the ST-BigData group for all the input and suggestions.

Secondly, I thank the NOVA University of Lisbon, the College of Sciences and Technologies and NOVA LINCS for their excellence in education and research.

I also intend to thank IPMA and Doctor Lourdes Bugalho for her cooperation. I would like to thank my parents and my brother, for their unwavering support, believing in me and providing with this opportunity.

I thank my loving girlfriend for all her help and patience.

Finally, I thank my friends and colleagues for all their advises, talks, coffee breaks and words of motivation.

ABSTRACT

Forest fires cause devastating amounts of damage generating negative consequences in the economy, the environment, the populations' quality of life and in worst case the loss of lives. Having this in mind, the quick and timely prediction of forest fires is a major factor in the mitigation or even negation of the aforementioned consequences.

Remote sensing is the process of obtaining information about an object or phenomena without direct interaction. This is the premise on which satellites acquire data of planet Earth. These observations produce enormous amounts of data on a daily basis. This data can be used to find correlation between land surface variables and conditions that are prone to fire ignition. Recently, in this field of study, there has been an effort to automate the process of correlation using machine learning techniques, such as Support Vector Machines and Artificial Neural Networks, in conjunction with a data mining approach, where historical data of a specific area is analysed in order to sort out the major primers of forest fire ignitions and identifying trends. The drawback of this approach is the large amount of time even the simplest task takes to process. GPU processing is the most recent strategy to accelerate this process.

The thesis aims to study the behaviour of GPU parallelized classifiers with the ever increasing amounts of data to process and understand if these are appropriate for use in forest predictive tasks.

Keywords: Remote Sensing, GPU Processing, Satellite Systems, Machine Learning, Support Vector Machines, Artificial Neural Networks, Gradient Boosting, K-Nearest Neighbours, Wildfires

RESUMO

Os fogos florestais são a causa de uma enorme devastação que são acompanhados de consequências económicas, ambientais, para a qualidade de vida das populações e em pior caso, a perda de vidas. Com isto em mente, a previsão rápida e atempada de fogos florestais é um fator importante na mitigação ou mesmo na prevenção em completo destas consequências.

Detecção Remota é o processo de obter informação sobre um objecto ou fenómeno sem recorrer a interação direta. Esta é a premissa sob a qual os satélites recolhem informação sobre o planeta Terra. Estas observações produzem, diariamente, enormes quantidades de dados. Estes dados podem ser correlacionados com variáveis da superfície terrestre e condições propícias para a deflagração de fogos. Nesta área de estudo, nos últimos anos, tem ocorrido um esforço para a automação do processo usando técnicas de aprendizagem automática, tais como Support Vector Machines e Random Forests, em conjunção com uma abordagem de Data Mining, onde dados históricos de uma determinada região são analisados de forma a encontrar os principais fatores na ignição de fogos florestais e a identificação de tendências. A contrapartida desta abordagem é o longo tempo que a mais simples tarefa pode demorar a processar face ao volume de dados. Esta dissertação terá como objectivo estudar o comportamento de classificadores aliados ao processamento paralelo em GPU quando enfrentados com uma quantidade incremental de dados e perceber se estes são apropriados para o uso em tarefas de previsão em florestas.

Palavras-chave: Remote Sensing, GPU Processing, Satellite Systems, Machine Learning, Support Vector Machines, Artificial Neural Networks, Gradient Boosting, K-Nearest Neighbours, Wildfires

CONTENTS

List of Figures	xv
List of Tables	xix
Listings	xxv
Acronyms	xxvii
1 Introduction	1
1.1 Motivation	1
1.2 Area of Study	3
1.3 Related Work	5
1.4 Problem Description	6
1.4.1 Execution Times	6
1.4.2 Land Cover Diversity	6
1.4.3 Resolution Diversity	7
1.5 Objectives	7
1.5.1 Burn Indice Analysis	7
1.5.2 CPU-bound vs GPU-accelerated execution	7
1.5.3 Satellite Comparison	8
1.5.4 Preliminary classification results	8
1.6 Contributions	8
1.7 Document Structure	8
2 Theoretical Basis	11
2.1 Remote Sensing	11
2.1.1 Remote Sensing Approaches	12
2.1.2 Satellite Systems	12
2.2 Classifiers	18
2.2.1 Classification Methods	18
2.2.2 Classifier Comparison Metrics	19
2.2.3 Classifiers	21
2.3 Parallel Processing	24

2.3.1	CPU + GPU Processing	25
2.3.2	Python + CUDA	25
2.3.3	Python Standard Library	26
2.3.4	Python GPU Accelerated Machine Learning Libraries	26
2.3.5	Libraries used	28
2.3.6	Benchmarking	28
2.3.7	Conclusion	28
3	Fire Indices and Fire Prediction Techniques	29
3.1	Spectral Signature	29
3.1.1	Burned Area Reflectance Classification	29
3.1.2	Normalized Difference Vegetation Index	31
3.1.3	Soil Adjusted Vegetation Index	31
3.1.4	Moisture Stress Index	32
3.1.5	Mid-infrared Burn Index	32
3.1.6	Normalized Burn Ratio	33
3.1.7	Deltas	34
3.1.8	Indice comparison	34
3.2	Canadian Fire Weather Index	37
3.3	Fire Prediction Techniques	38
3.3.1	Burned Area Identification	38
3.3.2	Fire Risk Classification	39
3.3.3	Summary	41
3.3.4	Conclusion	41
4	Experimental study	43
4.1	Methodology	43
4.1.1	Pre-Classification Phase	43
4.1.2	Classification Phase	43
4.1.3	Complementary Studies Phase	44
4.2	Raster processing	44
4.3	Feature Selection	44
4.4	Parameter Tuning	45
4.4.1	Multilayer Perceptron Classifier	45
4.4.2	Gradient Boosting Classifier	46
4.4.3	Support Vector Classifier	47
4.4.4	K Nearest Neighbours Classifier	47
4.5	Classification Process	48
4.6	Experimentation setting	48
4.6.1	Hardware specifications	49
4.7	Result Discussion	49

4.7.1	Organization of result discussion	49
4.7.2	Artificial Neural Networks	49
4.7.3	Gradient Boosting	52
4.7.4	Support Vector Machines	54
4.7.5	K Nearest Neighbours	56
4.7.6	Comparison	57
4.8	Complementary studies	63
4.8.1	Fire season burned area classification	63
4.8.2	Identification of areas with high risk of fire	64
4.8.3	Conclusion	66
5	Conclusion & Future Work	67
5.1	Conclusion	67
5.1.1	Classifiers	68
5.1.2	Satellites	68
5.2	Future Work	69
	Bibliography	71
A	Execution Times in Detail	77
A.1	Multilayer Perceptron Classifier	77
A.2	Deep Neural Network Classifier	79
A.3	Gradient Boosting Classifier	80
A.4	XGBoost Classifier	81
A.5	Support Vector Classifier	83
A.6	liquidSVM Classifier	84
A.7	K Nearest Neighbours Classifier	85
B	Classifier counterpart comparison	87
B.1	Neural Networks	87
B.2	Gradient Boosting	89
B.3	Support Vector Machines	90
C	Confusion Matrices for Burned Area Classification	93
C.1	Sentinel 2	93
C.1.1	Multilayer Perceptron Classifier	93
C.1.2	Deep Neural Network Classifier	94
C.1.3	Gradient Boosting Classifier	96
C.1.4	XGBoost Classifier	97
C.1.5	Support Vector Classifier	98
C.1.6	liquidSVM Classifier	99
C.1.7	K Nearest Neighbours Classifier	100

CONTENTS

C.2	Landsat 8	101
C.2.1	Multilayer Perceptron Classifier	101
C.2.2	Deep Neural Network Classifier	103
C.2.3	Gradient Boosting Classifier	104
C.2.4	XGBoost Classifier	105
C.2.5	Support Vector Classifier	106
C.2.6	liquidSVM Classifier	108
C.2.7	K Nearest Neighbours Classifier	109
C.3	MODIS	110
C.3.1	Multilayer Perceptron Classifier	110
C.3.2	Deep Neural Network Classifier	111
C.3.3	Gradient Boosting Classifier	113
C.3.4	XGBoost Classifier	114
C.3.5	Support Vector Classifier	115
C.3.6	liquidSVM Classifier	116
C.3.7	K Nearest Neighbours Classifier	117

LIST OF FIGURES

1.1	Thermopluviometric graph of Portugal based on averages of data from 1971 to 2000 - adapted from data present in the publication by IPMA and AEMet [19].	3
1.2	Instituto da Conservação da Natureza e das Florestas (ICNF)s burned area summary and area of study for 2016	4
2.1	Example of a Multi Layered Perceptron (MLP) with n inputs and one output class	21
2.2	K-Nearest Neighbours with k ranging from 1 to 5	22
2.3	Support Vector Machine	23
2.4	Gradient Boosted Trees	24
2.5	CPU vs GPU GFLOPS comparison over time	25
3.1	Spectral response for burned areas (BAER [5])	30
3.2	NDVI index and scale for 27 th of September using Sentinel 2 imagery.	31
3.3	SAVI index and scale for 27 th of September using Sentinel 2 imagery.	31
3.4	MSI index and scale for 27 th of September using Sentinel 2 imagery.	32
3.5	MIRBI index and scale for 27 th of September using Sentinel 2 imagery.	32
3.6	NBR index and scale for 27 th of September using Sentinel 2 imagery.	33
3.7	preview of the dNBR indice and scale for the interval between the 29 th of July and 27 th of September using Sentinel 2 imagery.	34
3.8	a) burned area classification confusion map using dNBR b) burned area classification confusion map using dNBR4 with Sentinel 2 imagery for September 27 th	35
3.9	Burned area classification confusion map using the RdNBR index with Sentinel 2 imagery for September 27 th	36
3.10	Schema of the Canadian Fire Weather Index	38
4.1	Graph of the non-zero LASSO score features	45
4.2	Multilayer Perceptron hidden_layer_sizes during Cross Validation	46
4.3	Gradient Boosting max_depth and min_samples_split during Cross Validation	46
4.4	Support Vector Machines Gamma parameter during Cross Validation	47
4.5	K Nearest Neighbours n_neighbors during Cross Validation	47

4.6	Diagram of the classification process	48
4.7	Total execution times of the MLP classifier across all satellite datasets and training set percentages	50
4.8	Total execution times of the DNN classifier across all satellite datasets and training set percentages	51
4.9	Total execution times of the GB classifier across all satellite datasets and training set percentages	52
4.10	Total execution times of the XGB classifier across all satellite datasets and training set percentages	53
4.11	Total execution times of the SVC classifier across all satellite datasets and training set percentages	54
4.12	Total execution times of the liquidSVM classifier across all satellite datasets and training set percentages	55
4.13	Total execution times of the KNN classifier across all satellite datasets and training set percentages	56
4.14	Overall accuracy.	57
4.15	Overall kappa.	57
4.16	Confusion maps generated from a) Sentinel 2 classification b) Landsat 8 classification c) MODIS classification.	59
4.17	a) Weighted Confusion Map	61
4.18	Weighted Confusion Map close-up.	62
4.19	a) shows the study area on the 27 th of September according to Sentinel 2 and b) shows classification result for 31 st of October.	63
4.20	a) shows ICNFs burned areas and b) shows the ground truth used.	65
4.21	a) shows the risk identification for August 10 th b) shows the risk identification for August 11 th . These classifications are overlayed on the ICNFs summary	66
A.1	Execution time detailing training and classification time for the MLP classifier using Sentinel 2 data	77
A.2	Execution time detailing training and classification time for the MLP classifier using Landsat 8 data	78
A.3	Execution time detailing training and classification time for the MLP classifier using MODIS data	78
A.4	Execution time detailing training and classification time for the DNN classifier using Sentinel 2 data	79
A.5	Execution time detailing training and classification time for the DNN classifier using Landsat 8 data	79
A.6	Execution time detailing training and classification time for the DNN classifier using MODIS data	79
A.7	Execution time detailing training and classification time for the GBC classifier using Sentinel 2 data	80

A.8 Execution time detailing training and classification time for the GBC classifier using Landsat 8 data	80
A.9 Execution time detailing training and classification time for the GBC classifier using MODIS data	81
A.10 Execution time detailing training and classification time for the XGB classifier using Sentinel 2 data	81
A.11 Execution time detailing training and classification time for the XGB classifier using Landsat 8 data	82
A.12 Execution time detailing training and classification time for the XGB classifier using MODIS data	82
A.13 Execution time detailing training and classification time for the SVC classifier using Sentinel 2 data	83
A.14 Execution time detailing training and classification time for the SVC classifier using Landsat 8 data	83
A.15 Execution time detailing training and classification time for the SVC classifier using MODIS data	83
A.16 Execution time detailing training and classification time for the liquidSVM classifier using Sentinel 2 data	84
A.17 Execution time detailing training and classification time for the liquidSVM classifier using Landsat 8 data	84
A.18 Execution time detailing training and classification time for the liquidSVM classifier using MODIS data	84
A.19 Execution time detailing training and classification time for the KNN classifier using Sentinel 2 data	85
A.20 Execution time detailing training and classification time for the KNN classifier using Landsat 8 data	85
A.21 Execution time detailing training and classification time for the KNN classifier using MODIS data	85
B.1 MLP classifier versus DNN classifier execution time comparison using Sentinel 2 data.	87
B.2 MLP classifier versus DNN classifier execution time comparison using Landsat 8 data.	88
B.3 MLP classifier versus DNN classifier execution time comparison using MODIS data.	88
B.4 GB classifier versus XGB classifier execution time comparison using Sentinel 2 data.	89
B.5 GB classifier versus XGB classifier execution time comparison using Landsat 8 data.	89
B.6 GB classifier versus XGB classifier execution time comparison using MODIS data.	90

B.7	SVC classifier versus liquidSVM classifier execution time comparison using Sentinel 2 data.	90
B.8	SVC classifier versus liquidSVM classifier execution time comparison using Landsat 8 data.	91
B.9	SVC classifier versus liquidSVM classifier execution time comparison using MODIS data.	91

LIST OF TABLES

1.1	Summary of the execution times in work from Nunes et al. [34].	6
2.1	Sentinel-2 Spectral Band specification (adapted from GDAL [13])	13
2.2	Landsat 8 Spectral Band details (USGS [50])	14
2.3	MODIS Spectral Bands (adapted from NASA [31])	16
2.4	Satellite Systems comparison	17
2.5	Classifier counterpart setup	27
3.1	Simplified Table of related works	41
4.1	Sample amounts per training set percentage.	48
4.2	Confusion Matrix from the MLP Classifier using 10% training data from Sentinel 2 imagery	50
4.3	Confusion Matrix from DNN Classifier using 30% training data from Sentinel 2 imagery	51
4.4	Confusion Matrix from GB Classifier using 10% training data from Sentinel 2 imagery	52
4.5	Confusion Matrix from XGB Classifier using 10% training data from Sentinel 2 imagery	53
4.6	Confusion Matrix from SVC classifier using 5% training data from Sentinel 2 imagery	54
4.7	Confusion Matrix from liquidSVM Classifier using 10% training data from Sentinel 2 imagery	55
4.8	Confusion Matrix from K Nearest Neighbours Classifier using 10% training data from Sentinel 2 imagery	56
4.9	Discretization of the voting outcome	61
4.10	Fire season classification metrics	63
4.11	List of features used for risk identification	64
C.1	Confusion Matrix from the MLP Classifier using 1% training data from Sentinel 2 imagery	93
C.2	Confusion Matrix from the MLP Classifier using 5% training data from Sentinel 2 imagery	93

C.3	Confusion Matrix from the MLP Classifier using 10% training data from Sentinel 2 imagery	94
C.4	Confusion Matrix from the MLP Classifier using 20% training data from Sentinel 2 imagery	94
C.5	Confusion Matrix from the MLP Classifier using 30% training data from Sentinel 2 imagery	94
C.6	Confusion Matrix from the DNN Classifier using 1% training data from Sentinel 2 imagery	94
C.7	Confusion Matrix from the DNN Classifier using 5% training data from Sentinel 2 imagery	95
C.8	Confusion Matrix from the DNN Classifier using 10% training data from Sentinel 2 imagery	95
C.9	Confusion Matrix from the DNN Classifier using 20% training data from Sentinel 2 imagery	95
C.10	Confusion Matrix from the DNN Classifier using 30% training data from Sentinel 2 imagery	95
C.11	Confusion Matrix from the GB Classifier using 1% training data from Sentinel 2 imagery	96
C.12	Confusion Matrix from the GB Classifier using 5% training data from Sentinel 2 imagery	96
C.13	Confusion Matrix from the GB Classifier using 10% training data from Sentinel 2 imagery	96
C.14	Confusion Matrix from the GB Classifier using 10% training data from Sentinel 2 imagery	96
C.15	Confusion Matrix from the GB Classifier using 10% training data from Sentinel 2 imagery	97
C.16	Confusion Matrix from the XGB Classifier using 1% training data from Sentinel 2 imagery	97
C.17	Confusion Matrix from the XGB Classifier using 5% training data from Sentinel 2 imagery	97
C.18	Confusion Matrix from the XGB Classifier using 10% training data from Sentinel 2 imagery	97
C.19	Confusion Matrix from the XGB Classifier using 10% training data from Sentinel 2 imagery	98
C.20	Confusion Matrix from the XGB Classifier using 10% training data from Sentinel 2 imagery	98
C.21	Confusion Matrix from the SVC Classifier using 1% training data from Sentinel 2 imagery	98
C.22	Confusion Matrix from the SVC Classifier using 5% training data from Sentinel 2 imagery	98

C.23 Confusion Matrix from the SVC Classifier using 10% training data from Sentinel 2 imagery	99
C.24 Confusion Matrix from the liquidSVM Classifier using 1% training data from Sentinel 2 imagery	99
C.25 Confusion Matrix from the liquidSVM Classifier using 5% training data from Sentinel 2 imagery	99
C.26 Confusion Matrix from the liquidSVM Classifier using 10% training data from Sentinel 2 imagery	99
C.27 Confusion Matrix from the liquidSVM Classifier using 10% training data from Sentinel 2 imagery	100
C.28 Confusion Matrix from the liquidSVM Classifier using 10% training data from Sentinel 2 imagery	100
C.29 Confusion Matrix from the KNN Classifier using 1% training data from Sentinel 2 imagery	100
C.30 Confusion Matrix from the KNN Classifier using 5% training data from Sentinel 2 imagery	100
C.31 Confusion Matrix from the KNN Classifier using 10% training data from Sentinel 2 imagery	101
C.32 Confusion Matrix from the KNN Classifier using 20% training data from Sentinel 2 imagery	101
C.33 Confusion Matrix from the KNN Classifier using 30% training data from Sentinel 2 imagery	101
C.34 Confusion Matrix from the MLP Classifier using 1% training data from Landsat 8 imagery	101
C.35 Confusion Matrix from the MLP Classifier using 5% training data from Landsat 8 imagery	102
C.36 Confusion Matrix from the MLP Classifier using 10% training data from Landsat 8 imagery	102
C.37 Confusion Matrix from the MLP Classifier using 20% training data from Landsat 8 imagery	102
C.38 Confusion Matrix from the MLP Classifier using 30% training data from Landsat 8 imagery	102
C.39 Confusion Matrix from the DNN Classifier using 1% training data from Landsat 8 imagery	103
C.40 Confusion Matrix from the DNN Classifier using 5% training data from Landsat 8 imagery	103
C.41 Confusion Matrix from the DNN Classifier using 10% training data from Landsat 8 imagery	103
C.42 Confusion Matrix from the DNN Classifier using 20% training data from Landsat 8 imagery	103

C.43 Confusion Matrix from the DNN Classifier using 30% training data from Landsat 8 imagery	104
C.44 Confusion Matrix from the GBC Classifier using 1% training data from Landsat 8 imagery	104
C.45 Confusion Matrix from the GBC Classifier using 5% training data from Landsat 8 imagery	104
C.46 Confusion Matrix from the GBC Classifier using 10% training data from Landsat 8 imagery	104
C.47 Confusion Matrix from the GBC Classifier using 10% training data from Landsat 8 imagery	105
C.48 Confusion Matrix from the GBC Classifier using 10% training data from Landsat 8 imagery	105
C.49 Confusion Matrix from the XGB Classifier using 1% training data from Landsat 8 imagery	105
C.50 Confusion Matrix from the XGB Classifier using 5% training data from Landsat 8 imagery	105
C.51 Confusion Matrix from the XGB Classifier using 10% training data from Landsat 8 imagery	106
C.52 Confusion Matrix from the XGB Classifier using 10% training data from Landsat 8 imagery	106
C.53 Confusion Matrix from the XGB Classifier using 10% training data from Landsat 8 imagery	106
C.54 Confusion Matrix from the SVC Classifier using 1% training data from Landsat 8 imagery	106
C.55 Confusion Matrix from the SVC Classifier using 5% training data from Landsat 8 imagery	107
C.56 Confusion Matrix from the SVC Classifier using 10% training data from Landsat 8 imagery	107
C.57 Confusion Matrix from the SVC Classifier using 10% training data from Landsat 8 imagery	107
C.58 Confusion Matrix from the SVC Classifier using 10% training data from Landsat 8 imagery	107
C.59 Confusion Matrix from the liquidSVM Classifier using 1% training data from Landsat 8 imagery	108
C.60 Confusion Matrix from the liquidSVM Classifier using 5% training data from Landsat 8 imagery	108
C.61 Confusion Matrix from the liquidSVM Classifier using 10% training data from Landsat 8 imagery	108
C.62 Confusion Matrix from the liquidSVM Classifier using 20% training data from Landsat 8 imagery	108

C.63 Confusion Matrix from the liquidSVM Classifier using 30% training data from Landsat 8 imagery	109
C.64 Confusion Matrix from the KNN Classifier using 1% training data from Landsat 8 imagery	109
C.65 Confusion Matrix from the KNN Classifier using 5% training data from Landsat 8 imagery	109
C.66 Confusion Matrix from the KNN Classifier using 10% training data from Landsat 8 imagery	109
C.67 Confusion Matrix from the KNN Classifier using 20% training data from Landsat 8 imagery	110
C.68 Confusion Matrix from the KNN Classifier using 30% training data from Landsat 8 imagery	110
C.69 Confusion Matrix from the MLP Classifier using 1% training data from MODIS imagery	110
C.70 Confusion Matrix from the MLP Classifier using 5% training data from MODIS imagery	110
C.71 Confusion Matrix from the MLP Classifier using 10% training data from MODIS imagery	111
C.72 Confusion Matrix from the MLP Classifier using 20% training data from MODIS imagery	111
C.73 Confusion Matrix from the MLP Classifier using 30% training data from MODIS imagery	111
C.74 Confusion Matrix from the DNN Classifier using 1% training data from MODIS imagery	111
C.75 Confusion Matrix from the DNN Classifier using 5% training data from MODIS imagery	112
C.76 Confusion Matrix from the DNN Classifier using 10% training data from MODIS imagery	112
C.77 Confusion Matrix from the DNN Classifier using 20% training data from MODIS imagery	112
C.78 Confusion Matrix from the DNN Classifier using 30% training data from MODIS imagery	112
C.79 Confusion Matrix from the GBC Classifier using 1% training data from MODIS imagery	113
C.80 Confusion Matrix from the GBC Classifier using 5% training data from MODIS imagery	113
C.81 Confusion Matrix from the GBC Classifier using 10% training data from MODIS imagery	113
C.82 Confusion Matrix from the GBC Classifier using 10% training data from MODIS imagery	113

C.83 Confusion Matrix from the GBC Classifier using 10% training data from MODIS imagery	114
C.84 Confusion Matrix from the XGB Classifier using 5% training data from MODIS imagery	114
C.85 Confusion Matrix from the XGB Classifier using 10% training data from MODIS imagery	114
C.86 Confusion Matrix from the XGB Classifier using 10% training data from MODIS imagery	114
C.87 Confusion Matrix from the XGB Classifier using 10% training data from MODIS imagery	115
C.88 Confusion Matrix from the SVC Classifier using 1% training data from MODIS imagery	115
C.89 Confusion Matrix from the SVC Classifier using 5% training data from MODIS imagery	115
C.90 Confusion Matrix from the SVC Classifier using 10% training data from MODIS imagery	115
C.91 Confusion Matrix from the SVC Classifier using 10% training data from MODIS imagery	116
C.92 Confusion Matrix from the SVC Classifier using 10% training data from MODIS imagery	116
C.93 Confusion Matrix from the liquidSVM Classifier using 1% training data from MODIS imagery	116
C.94 Confusion Matrix from the liquidSVM Classifier using 5% training data from MODIS imagery	116
C.95 Confusion Matrix from the liquidSVM Classifier using 10% training data from MODIS imagery	117
C.96 Confusion Matrix from the liquidSVM Classifier using 10% training data from MODIS imagery	117
C.97 Confusion Matrix from the liquidSVM Classifier using 10% training data from MODIS imagery	117
C.98 Confusion Matrix from the KNN Classifier using 1% training data from MODIS imagery	117
C.99 Confusion Matrix from the KNN Classifier using 5% training data from MODIS imagery	118
C.100 Confusion Matrix from the KNN Classifier using 10% training data from MODIS imagery	118
C.101 Confusion Matrix from the KNN Classifier using 10% training data from MODIS imagery	118
C.102 Confusion Matrix from the KNN Classifier using 10% training data from MODIS imagery	118

LISTINGS

4.1	Voting algorithm	60
4.2	Fire Weather Indice geographical referencing algorithm	64

ACRONYMS

ANN	Artificial Neural Networks.
ANPC	Autoridade Nacional de Proteção Civil.
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer.
BARC	Burned Area Reflectance Classification.
BR	Burn Ratio.
BUI	Buildup Index.
CERES	Clouds and Earth's Radiant Energy System.
CPU's	Central Processing Unit.
CSV	Comma Sepparated Values.
CUDA	Compute Unified Device Architecture.
DC	Drought Code.
DMC	Duff Moisture Code.
DOS1	Dark Object Subtraction 1.
DSR	Daily Severity Rating.
EEA	European Environment Agency.
EFFIS	European Forest Fire Information System.

ACRONYMS

EOS	Earth Observing System.
EOSPSO	EOS Project Science Office.
ESA	European Space Agency.
FFMC	Fine Fuel Moisture Code.
FN	False Negative.
FP	False Positive.
FWI	Fire Weather Index.
GB	Gradient Boosting.
GFLOPS	Giga Floating-point Operation per Second.
GPU	Graphics Processing Unit.
ICNF	Instituto da Conservação da Natureza e das Florestas.
IPMA	Instituto Português do Mar e da Atmosfera.
ISI	Initial Spread Index.
KNN	K Nearest Neighbours.
LSWIR	Long Shortwave Infrared.
LWIR	Longwave Infrared.
MIRBI	Mid-infrared Burn Index.
MISR	Multi-angle Imaging SpectroRadiometer.

ML	Machine Learning.
MLP	Multi Layered Perceptron.
MODIS	Moderate-resolution Imaging Spectroradiometer.
MOPITT	Measurements of Pollution in the Troposphere.
MSI	Multi Spectral Instrument.
MSI	Moisture Stress Index.
MWIR	Midwave Infrared.
NASA	National Aeronautics and Space Administration.
NBR	Normalized Burn Index.
NDVI	Normalized Difference Vegetation Index.
NIR	Near Infrared Spectrum.
NRCan	Natural Resources Canada.
OLI	Operational Land Imager.
RBF	Radial Basis Function.
RGB	Red Green Blue.
SAVI	Soil Adjusted Vegetation Index.
SIMD	Single Instruction Multiple Data.
SSWIR	Short Shortwave Infrared.
SVM	Support Vector Machine.

ACRONYMS

SWIR Shortwave Infrared Spectrum.

TIR Thermal Infrared.

TIRS Thermal Infrared Sensor.

TN True Negative.

TP True Positive.

USGS U.S. Geological Survey.

VIS Visible Spectrum.

INTRODUCTION

According to [Instituto Português do Mar e da Atmosfera \(IPMA\)](#) [18], Portugal's forest fire season starts on May and continues on till October, which related to the lowest values of precipitation and highest temperatures of the year, as seen in Figure 1.1. During this season forest fires take their toll on Portugal, making millions of euros in damages and burning thousands of hectares of forest and property. Portugal, in spite its size, features the largest amounts of yearly fires and follows Italy by a small margin in the extent of burned area [7].

In the yearly report by [ICNF](#) [16] in 2015, where it states the various facts about that year's fire season, it states that Portugal lost almost 130 million euros in losses and extraordinary expenses derived from forest fires. This hinders Portugal's economy, tourism and quality of life in addition leaving a scar across the landscape. Recently, during the fire season of 2017, Portugal saw one of the worst recorded wildfire calamities in its history with the examples of the great fires that occurred in the Pedrógão Grande region and Pinhal de Leiria, one of the largest natural reserves in the country. [ICNF](#) [4] reports that 442.418 ha have burned during 2017 resulting in a 428% increase when comparing to the previously recorded average.

1.1 Motivation

For the intent of managing natural resources, there have been efforts in the area of burned area identification and wildfire risk prediction. These tasks have been used in association with machine learning algorithms in order to achieve better results through automated processes, but there are some caveats associated with these combinations. Since these tasks have big datasets the algorithms take long amounts of time to execute. Consequently wildfire related tasks using remotely sensed data can benefit from [Graphics Processing](#)

Unit (GPU) processing in many areas, due to the GPUs enormous computational capability. These intensive tasks, for example the identification of burned areas in a timely fashion, can make all the difference in the management of fire fighting personnel and assets on the field optimizing their effectiveness. The applications we suggest as being suitable for the use of this technology are:

- **Burned Area Mapping** - The rapid classification of the burned areas can result a decrease in the amount of work and manpower using specialized personnel on the ground. For example, the use of GPUs associated with Machine Learning (ML) algorithms could speed up the process of classifying burned areas in addition to possibly better identification results when processing such huge amount of data. By having faster algorithms for this task, resources can be made available more frequently and used for numerous purposes.
- **Wildfire Risk Identification** - Wildfire is the result multivariate combination of the time of the year, the terrain type and the meteorological condition. The assessment of this risk is as such, a data heavy task which is computation intensive, resource demanding and time consuming even for the execution of a simple task. This hinders progress and tool development. Hardware acceleration is the new trend for the solution of these problems and surpass the current processing bottleneck associated with the execution of such high dimensionality problems.

Remote Sensing enabled the collection of data from dangerous and inaccessible areas, it also replaces the need for the gathering of data on the ground. Satellite Imagery is used in numerous purposes, for example, detection of landscape structure and precise estimates of leaf and biomass indices. With the constant improvement of satellite systems, their sophisticated sensors can now generate imagery of great quality at numerous resolutions and scales, providing with high quality data that can be used to estimate the Forested areas conditions. The multi-temporality of these observations, are essential for applications since it enables que compilation of a timeline of events and changes in remotely sensed areas. The diversity of wavelengths captured and converted into images provide new information at resolution that isn't available by traditional means. It is our intention to use Sentinel 2, Landsat 8 and Moderate-resolution Imaging Spectroradiometer (MODIS) as our sources for remotely sensed imagery. This is derived from the image quality associated with Sentinel 2 and Landsat 8 as well as MODIS' daily coverage, in addition to the products generated by these satellites being available to the public in general.

Processing these large amounts of data requires machines with tremendous throughput capabilities. But these machines come with a big price tag attached, whereas in the hardware cost itself, as in the it maintenance cost. After Central Processing Units (CPUs) having reached somewhat of a wall, where the computational power gained with each iteration of the architecture is not as substantial as it once was. Attentions start to

turn to GPUs, which comparatively provide ever growing potential for integration with the processing of hardware demanding problems, at a more affordable price per Giga Floating-point Operation per Second (GFLOPS). This has been the highlight of NVIDIA's deep learning segment. Still, there are some caveats that attached to this sort of approach, which will be further explained in Chapter 2.3.

In short, the remote sensing and forest fire prediction effort can benefit from the use of new technologies in order to generate a new generation of platforms that separate themselves from the computational specifications of big processing clusters, embracing new designs that use the silicon on the hardware in a more efficient and optimized manner.

1.2 Area of Study

Portugal is a country with 92.212 Km^2 (9221200 ha) situated on the Iberian Peninsula, in south-eastern Europe. It is bathed by the Atlantic to the east and Spain to the west and that is divided into 18 districts, plus the two autonomous regions of Madeira and Azores. In north the landscape is characterized by mountainous terrain and in the interior areas with plateaus. The south, on the other hand, as far as the Algarve features mostly rolling plains and a mountainous ridge that spans it's northern border. The climate in Portugal is characterised as being a Tempered Mediterranean Climate, experiencing a wet season during the winter and a dry season during the summer.

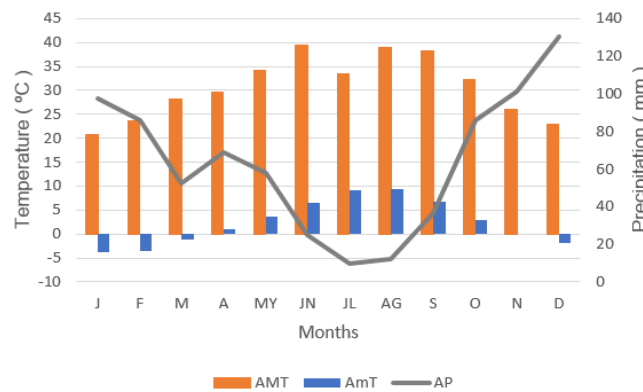


Figure 1.1: Thermopluviometric graph of Portugal based on averages of data from 1971 to 2000 - adapted from data present in the publication by IPMA and AEMet [19].

J-January, F-February, M-March, A-April, MY-May, JN-June, JL-July, AG-August, S-September, O-October, N-November, D-December, AMT-Average Maximum Temperature, AmT-Average Minimum Temperature, AP-Average Precipitation.

Based on analysing yearly burned area summary released by ICNF [15], the northern

regions of Portugal has a greater fire occurrence density, therefore providing a larger possibility for datasets to be used in this study. The north of Portugal is characterized by large woodlands with three predominant species (Oaks, Eucalyptus and Pine trees). The terrain is comprised of mountainous ridges and hills. It has a Mediterranean climate usually characterized by rainy winters and dry, warm summers. The annual average rainfall varies around 3,000 mm. The temperatures vary from -3°C to 38°C , with an annual average of about 13°C .

The area chosen for our study has 559KM^2 and is situated in the Castelo de Paiva region. This region saw significant fire activity during the summer months of 2016, which resulted in a 118 Km^2 , nearly $\frac{1}{5}$ of our total study area. This area is characterized by two intersecting rivers (Tâmega and Douro) as well as some towns, quarries and other man-made structures. Figure 1.2 illustrates the burned area summary for 2016 with the selected area of study in detail.

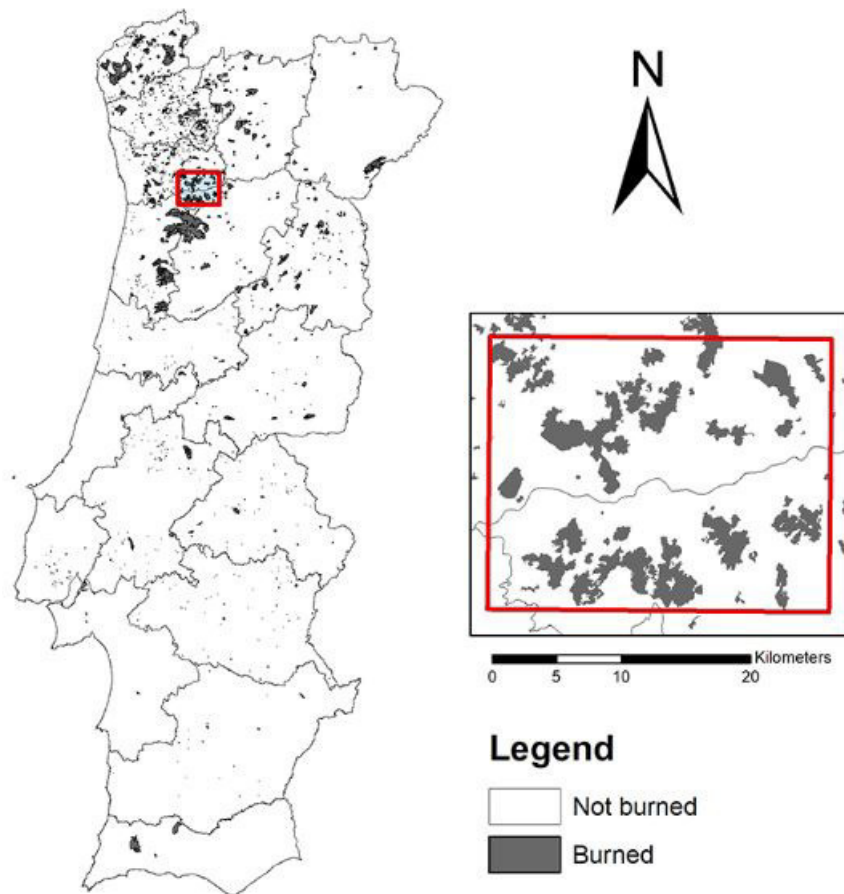


Figure 1.2: ICNFs burned area summary and area of study for 2016

1.3 Related Work

Investigators from the Dom Luiz Institute of the University of Lisbon, with support from the Superior Institute of Agronomy (UTAD) and [ICNF](#), have created a platform which enabled the consulting of a fire risk map forecast for the current day and the next couple of days. This map is generated based on meteorological observations, vegetation stress index and Canadian Fire Weather Index components. The whole system has statistical basis where it compares the forecasts to historical averages, calculates the risk based on the compared values and overlays the predictions over satellite imagery. This platform also pinpoints on the map active fire scenarios report by the [Autoridade Nacional de Proteção Civil \(ANPC\)](#).

The ESA-backed, Earth Observation-based Risk-EOS burn scar mapping service began in 2004. It uses satellite imagery from SPOT and Landsat to automatically detect burn scars. Burn scar detection is planned to take place on a seasonal basis, identifying fires covering at least one hectare to a standard resolution of 30 metres, with detailed damage assessment available to a maximum resolution of 2.5 metres using the SPOT 5 satellite. This service is already being used by Italy's National Civil Protection Department, Spain's Dirección general para la Biodiversidad and France's National Department of Civil Protection and Centre D'Essais Et De Recherche de l'Entente. Italy's National Civil Protection Department is providing advice on the implementation of the Risk-EOS service, based on previous experience with an ESA Data User Programme (DUP) project called ITALSCAR. This project was used to chart burn scars across the whole of Italian territory occurring between June and September during the years 1997, 1998, 1999 and 2000. The methodology used by ITALSCAR consisted on the mapping of burn scars pixel by pixel using an automated software system, followed up with manual photo-interpretation for quality assurance [44]. [European Forest Fire Information System \(EFFIS\)](#) consists on web geographic information system that provides near-real time historical information on forest fires and their regimes. This service encompasses the European, Middle Eastern and North African regions. [EFFIS](#) is composed of five modules, Fire Danger Assessment, Rapid Damage Assessment (these last two have near-real time availability), Emissions Assessment and Smoke Dispersion, Potential Soil Loss Assessment, and Vegetation Regeneration. In summary, [EFFIS](#) is a Fire Database for 22 countries and since 2015, it became one of the components of the Emergency Management Services in the European Union Copernicus program [10, 11].

1.4 Problem Description

This section has to layout the problems this study intends to address and are as follows:

1.4.1 Execution Times

The use of machine learning algorithms in past studies made at NOVA University, for example Nunes et al. [34] has shown that the assessment ML algorithms with remotely sensed data for the task of land cover identification have achieved good results. The only problem is that some algorithms tend to take a large amount of time to complete their execution as the dataset size increases. This poses an obstacle in research processes of this nature. Table 1.4.1 summarizes the execution times from the classifiers used in their work.

Table 1.1: Summary of the execution times in work from Nunes et al. [34].

Classifier	Time (s)	
	Minimum	Maximum
Support Vector Machine	7332.75	15210.83
Random Forest	190.5	264.03
K Nearest Neighbours	668.91	3441.45
Decision Trees	33.55	52.43
Maximum Likelihood	668.91	3441.45

Some of the classifiers used possess a high execution time, and may have potential for acceleration if some of the algorithms heavy computations can be converted to parallel execution. The [Support Vector Machine \(SVM\)](#), [K Nearest Neighbours \(KNN\)](#) and [Maximum Likelihood](#) pose as good candidates due to the high upper-bounds on the overall execution times.

1.4.2 Land Cover Diversity

Mapping complex landscapes over large areas containing such a wide range of contents can prove a particularly challenging task. Although a the binary classification of burned areas may have a simple result, the process of rooting out elements that possess nearly identical pixel values (like rock formations) but are not related to intended tasks objectives can pose some difficulty if done visually. So it is expected that the accuracy of the classification may differ from the pre-established classification provided by [ICNF](#). Yet, the use of imagery from both before and after the wildfire activity occurring may highlight useful patterns in the change oh the landscape, improving the accuracy of the classification.

1.4.3 Resolution Diversity

Another problem that was faced was the different resolutions from Landsat 8, Sentinel 2 and MODIS. In our area of study, burned areas come in different shapes and sizes, there can be areas that be either in thousands of square meters (m^2) or only a few tens. This can mean a huge difference when comparing for example, MODIS imagery which has 250 m resolution (62500 m^2 per pixel), now lets consider a hypothetical burned area that only has 600 m^2 , this can lead to the blending or even the obscuring of this burned area using MODIS imagery since its spectral signature might not be sufficiently pronounced or mixed with too much noise. These conditions may prove disastrous in the training phase, since the pixel size is so large, the number of samples corresponding to the area of study can be too few for a correct classification, or even worse, cause some irregularities in the algorithms executions.

1.5 Objectives

The objective is to perform a machine learning classifier benchmark of both [CPU's](#)-bound and [GPU](#)-accelerated using data derived from Sentinel 2, Landsat 8 and MODIS remotely sensed imagery from the Castelo de Paiva Region. Then compare their performance and effectiveness on the task of performing a binary burned area identification. We also aim to discover what are the best feature combinations for land cover identifications related to wildfire activities such as burned area identification.

The next subsections provide a small description of the main objectives.

1.5.1 Burn Indice Analysis

In order to analyse which features were relevant to burned area identification we decided to compare some of standard indices used for this task in addition to some variations of those indices and .

1.5.2 CPU-bound vs GPU-accelerated execution

Another objective and the most important one probably is to identify the best libraries that support machine learning tasks in [GPUs](#) in order to assert if that library proves to be good alternative to its [CPU's](#) version for the delegated task. This enables us to ponder the feasibility of their use in future endeavours over their [CPU's](#) counterparts.

For this effort it is our intent to perform the benchmark of those implementations for wildfire related tasks, with a varying amount dataset sources and training set percentages. This will provide a broad insight at the diverse strengths and weaknesses of each implementation.

1.5.3 Satellite Comparison

Another main objective is to compare how the images originating from different satellite missions affect each classification process. The missions chosen for this purpose were the Sentinel 2, Landsat 8 and Terra (MODIS).

1.5.4 Preliminary classification results

With the progress of this study it is our intention to generate preliminary classification and metrics for the wildfire related tasks, in this case burned area mapping and wildfire risk identification, using available ground-truth data for Portugal. For this task we intend to use [Artificial Neural Networks \(ANNs\)](#), [Gradient Boosting \(GB\)](#), [SVMs](#) and [KNN](#) classifiers.

1.6 Contributions

This study sets its aim on contributing to the field Burned Area Identification, and in a more generalized way to the fields relating to land cover classification and decision support systems. This while performing a comparison of [GPU](#) and [CPU's](#)-based classifier performance in wildfire related tasks. It is also intended to perform an analysis of the variation in performance when paired with Sentinel 2, Landsat 8 and [MODIS](#) imagery as well as a suggestion of which remote sensing features are relevant for the task of burned area identification. In sum, it is our intent to contribute to:

- The fields of automatic land cover classification;
- With a comparison of how different data sources influence classifier performance
- With an in depth comparison of how [CPU's](#) performs against [GPU](#)-accelerated classifiers.

1.7 Document Structure

The following order was thought out with the intent of providing an organized and pleasant reading of the document. The document starts with the current chapter, the Introduction, in order to set a background for the study, as well as the motivation that promoted it.

In Chapter 2 delves into the subject of processing, explaining the particularities of both [CPU's](#) and [GPU](#) processing and their relation with the Python programming language.

Chapter 3 intends to give an insight into the domain of machine learning and the classifiers mentioned in chapter two. The metrics used to evaluate and compare these classifiers are also explain in this chapter.

Chapter 4 explains the diverse concepts of remote sensing and indices generated from remotely sensed data, along side the satellite system that generate these remote sensed observations.

In Chapter 5 the aim is set on explaining what is the standard for wildfire risk forecast in Portugal in addition to the state-of-the-art in the fields this study intends to cover, namely burned area mapping and wildfire risk prediction, using a machine learning approach.

THEORETICAL BASIS

This chapter has the goal of explaining the basic concepts for the subjects covered during this study, in order to establish a basic understanding of the concepts used in later chapters.

2.1 Remote Sensing

Remote Sensing is the science and the art of obtaining information, measurement or acquisition, through the analysis of data acquired without a direct contact with the object, area, or phenomenon under investigation (Lillesand and Kiefer 1987). Identification of land cover establishes the baseline from which monitoring activities (change detection) can be performed, and provides the ground cover information for baseline thematic maps.

Land Cover according to [Natural Resources Canada \(NRCan\)](#), refers to the surface cover on the ground, whether vegetation, urban infrastructure, water, bare soil or other. Identifying, delineating and mapping land cover is important for global monitoring studies, resource management, and planning activities. Identification of land cover establishes the baseline from which monitoring activities (change detection) can be performed, and provides the ground cover information for baseline thematic maps [9].

For regional mapping, continuous spatial coverage over large areas is required. It would be difficult to detect regional trends with point source data. Remote sensing fulfills this requirement, as well as providing multispectral, multisource, and multitemporal information for an accurate classification of land cover.

2.1.1 Remote Sensing Approaches

Remote sensing can be achieved by one of two ways. The first one is defined as passive sensing. In passive sensing the sensor is designed to receive and measure natural emissions produced by the Earth's surface and atmosphere's constituents. These sensors measure the surface composition, physical temperature among other characteristics of the Earth through a measured power function. The band frequencies used for sensing are determined by the physical properties of what is intended to measure. These properties do not change over time and the information acquired cannot be duplicated by other band frequencies. The second approach is active sensing, requires the sensor to measure signals that are reflected, refracted or scattered by the Earth's surface, atmosphere or any of its constituents. These sensors are applied to meteorology (measurement of rainfall, cloud profiles, etc) or the observation of the Earth's surface[30].

2.1.2 Satellite Systems

Satellites are usually defined as natural space bodies that orbit around another space body, just as the moon is a satellite of Earth and Earth is a satellite of the Sun. Currently, this definition is also applied to artificial spacecrafts which are put into orbit using rocket and held there by means of gravitational forces. Satellites are highly specialized wireless receivers/transmitters which have the main function of relaying radio-frequency waves and its encoded information from one corner of the world to another. Currently hundreds of satellites are operation around the planet [8]. Remote Sensing Satellites are equipped with an instrument (Radiometer) that quantitatively measures the intensity of electromagnetic radiation in some bands within the spectrum. This instrument can be further identified by the spectrum portion it covers. Examples of coverage are:

- **Visible Spectrum (VIS)** that can be divided into the **Red Green Blue (RGB)** wavelengths;
- **Infrared Spectrum** that is composed by:
 - **Near Infrared Spectrum (NIR)**;
 - **Shortwave Infrared Spectrum (SWIR)** which can be further specified into the **Short Shortwave Infrared (SSWIR)** and **Long Shortwave Infrared (LSWIR)** wavelengths;
 - **Midwave Infrared (MWIR)**;
 - **Longwave Infrared (LWIR)** which are used for temperature measurement and commonly denominated as **Thermal Infrared (TIR)**
- **Microwave Spectrum.**

2.1.2.1 COPERNICUS Programme

Copernicus is the new name for the Global Monitoring for Environment and Security programme.

[European Space Agency \(ESA\)](#) is developing a new family of satellites, called Sentinels, specifically for the operational needs of the Copernicus programme.

Each Sentinel mission is based on a constellation of two satellites to fulfil revisit and coverage requirements, providing robust datasets for Copernicus Services[1].

Sentinel 2

The Sentinel-2 mission is composed by twin satellites that fly on the same Sun-synchronous orbit, phased at 180° of each other. These satellites are Sentinel 2A and Sentinel 2B, which were launched on June 2015 and March 2017 respectively, with Sentinel 2C expected to launch in 2021 and Sentinel 2D in the same decade. This mission monitors the land surface conditions through a combination of the large swath width of 290 Km, spectral range and with high-revisit frequency (5 day intervals at the equator). The limits for coverage are 84° north and 84° south [12].

With its 13 spectral bands, 290 Km swath width and high revisit frequency, Sentinel 2's [Multi Spectral Instrument \(MSI\)](#) reduces the time required to build a European cloud-free image archive. The spectral bands of Sentinel 2 will provide data for land cover/change classification, atmospheric correction, cloud/snow separation and vegetation [1].

Table 2.1: Sentinel-2 Spectral Band specification (adapted from GDAL [13])

Band	Bandwidth (nm)	Resolution (m)
1 – Coastal Aerosol	0.433 – 0.453	60
2 – Blue	0.457 – 0.523	10
3 – Green	0.542 – 0.578	10
4 – Red	0.650 – 0.680	10
5 – Veg. Red Edge	0.697 – 0.713	20
6 – Veg. Red Edge	0.732 – 0.748	20
7 – Veg. Red Edge	0.773 – 0.793	20
8 – NIR	0.784 – 0.900	10
8a – Veg. Red Edge	0.855 – 0.875	20
9 – Water Vapour	0.935 – 0.955	60
10 – SWIR Cirrus	1.370 – 1.390	60
11 – SSWIR	1.565 – 1.655	20
12 – LSWIR	2.100 – 2.280	20

2.1.2.2 Landsat Programme

Landsat is a joint effort of the [U.S. Geological Survey \(USGS\)](#) and [National Aeronautics and Space Administration \(NASA\)](#).

The [USGS](#) delivers high quality systematic, geometric, radiometric, and terrain corrected data, which records natural and human-induced changes on the global landscape, the providing it to users worldwide [51].

Landsat 8

Landsat 8 orbits the Earth at an altitude of 705 Km in a 185 Km swath, moving from north to south over the sunlit side of the Earth in a sun synchronous orbit. The satellite makes a complete orbit every 99 minutes, completes about 14 full orbits each day, and crosses every point on Earth once every 16 days. It was launched as the Landsat Data Continuity Mission on February 11, 2013, contains the [Operational Land Imager \(OLI\)](#) and the [Thermal Infrared Sensor \(TIRS\)](#). OLI collects data with a spatial resolution of 30 meters in the [VIS](#), [NIR](#), and [SWIR](#) wavelength regions, and a 15-meter panchromatic band, which provides data compatible with products from previous missions. OLI also contains a deep blue band for coastal-aerosol studies and a band for cirrus cloud detection. The TIRS contains two thermal bands, in order to enable monitoring of surface temperature with two spectral windows.

Landsat data supports a vast range of applications in areas such as global change research, agriculture, forestry, geology, land cover mapping, resource management, water, and coastal studies. Specific environmental monitoring activities such as deforestation research, volcanic flow studies, and understanding the effects of natural disasters all benefit from the availability of Landsat data [51].

Table 2.2: Landsat 8 Spectral Band details (USGS [50])

Band	Bandwidth (um)	Resolution (m)
1 – Coastal Aerosol	0.433 – 0.453	30
2 – Blue	0.450 – 0.515	30
3 – Green	0.525 – 0.600	30
4 – Red	0.630 – 0.680	30
5 – NIR	0.845 – 0.885	30
6 – SSWIR	1.560 – 1.660	30
7 – LSWIR	2.100 – 2.300	30
8 – PAN	0.500 – 0.680	15
9 - Cirrus	1.360 – 1.390	30
10 – TIR 1	10.30 – 11.30	100
11 – TIR 2	11.50 – 12.50	100

2.1.2.3 Earth Observing System Project

NASA's Earth Observing System (EOS) is a coordinated series of polar-orbiting and low inclination satellites for long-term global observations of the land surface, biosphere, solid Earth, atmosphere, and oceans. The EOS Project Science Office (EOSPSO) is committed to bringing program information and resources to the Earth science research community and the general public alike.

Terra

EOS AM-1, nicknamed Terra, was launched on December 1999 and is part of a constellation of two distinct satellites. Terra, which we will overview here, its sister satellite is Aqua, which is tasked to observe precipitation, evaporation and the water cycle. Terra's observations taken together, provide unique insight into how the Earth system works and how it is changing. Terra observations reveal humanity's impact on the planet and provide crucial data about natural hazards like fire and volcanoes. These observations generate data about the Earth's bio-geochemical and energy systems by using five sensors that observe the atmosphere, land surface, oceans, snow and ice, and energy budget. Each sensor has unique features that enable scientists to meet a wide range of science objectives. The five Terra onboard sensors are Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), Clouds and Earth's Radiant Energy System (CERES), Multi-angle Imaging SpectroRadiometer (MISR), MODIS and Measurements of Pollution in the Troposphere (MOPITT).

From this assortment of instruments we will look at MODIS with greater detail. MODIS is a multi-spectral cross-track scanning radiometer that operates from the visible through the thermal infrared electromagnetic spectrum. A multidisciplinary instrument, designed to measure high-priority features (atmospheric, land surface, among other). The instrument operates in 36 spectral bands: Two of the bands have 250m resolution, five have 500m resolution, and twenty-nine bands have 1Km resolution. A complete listing of these bands can be seen in Table 2.3. MODIS has a large swath width of 2300Km, giving it the capability to cover the entire globe every 1 to 2 days. Wide spectral coverage and a good repeat cycle give MODIS the edge it needs to monitor so many different global parameters.

Because all five instruments are on the same satellite making simultaneous observations, scientists are able to compare different aspects of Earth's characteristics over time [25][31]. During this study, whenever we refer to MODIS we are referring to EOS Terra using this instrument. This satellite generates a wide range of products based on its numerous bands. For the purpose of this study we opted for the MOD09GA which provides surface daily reflectance. This products band summary is presented in 2.3

Table 2.3: MODIS Spectral Bands (adapted from NASA [31])

Band	Bandwidth (nm)	Resolution (m)
1 - Red	0.620 – 0.670	250
2 – NIR 1	0.841 – 0.876	250
3 – Blue	0.459 – 0.479	500
4 – Green	0.545 – 0.565	500
5 – NIR 2	1.230 – 1.250	500
6 – SSWIR	1.628 – 1.652	500
7 – LSWIR	2.105 – 2.155	1000

2.1.2.4 Satellite Comparison

Table 2.4: Satellite Systems comparison

Satellites	Sentinel-2	Landsat 8	Terra
Launch date	June 2015 / March 2017	February 2013	December 1999
Bands	VIS, NIR, SWIR	VIS, NIR, SWIR, TIR	VIS, NIR, SWIR
Number of bands	13	11	36
Spatial resolution	10 m (VIS & NIR) 20 m (SWIR)	30 m (VIS, NIR, SWIR) 100 m (TIR)	250 m (bands 1 - 2) 500 m (bands 3 - 7) 1000 m (bands 8 - 36)
Area per pixel	100 m ² (VIS & NIR) 400 m ² (SWIR)	900 m ² (VIS, NIR, SWIR) 10 Km ² (TIR)	62.5 Km ² (bands 1 - 2) 250 Km ² (bands 3 - 7) 1000 Km ² (bands 8 - 36)
Nominal orbit	786 Km	705 Km	705 Km
Polar orbit	Yes	No	Yes
Sun-synchronous	Yes	Yes	Yes
Temporal resolution	5 days	16 days	1 day

The temporal resolution specifies how long the satellite takes to complete a full scan of the planet. Terra's temporal resolution only specifies the time the MODIS instrument takes to fully scan the globe.

2.2 Classifiers

In this chapter we set our goals on explaining the basic concepts of machine learning and the classifiers chosen to be used in this study in addition to the metrics used for classifier assessment. The classifiers chosen were based on the ones present in the literature (see Section 3.3).

2.2.1 Classification Methods

Classifiers can be divided into two distinct categories, those whose classification is supervised and the ones who are unsupervised. Both these categories can be even further divided into classifiers whose learning process can be either eager or lazy. This section aims to give a simplified overview of these aspects.

2.2.1.1 Supervised classification

Supervised classification is the machine learning task that consists on inferring a mathematical function from labelled training data. The training data is composed of a set of training examples. In supervised learning, each example is a pair of an attribute and its corresponding class. A supervised learning algorithm analyses the training data and generates an inferred mathematical function, this is defined as fitting or training. The inferred function is then used to attribute a class to new examples. In a perfect scenario the classifier accurately classifies every new example to its corresponding class. For this, the learning algorithm requires a generalization based on training data for unseen situations in a "reasonable" way (inductive bias) [2][34].

2.2.1.2 Unsupervised classification

Unsupervised classification is the machine learning task that consists on inferring a mathematical function that describes a hidden structure from unlabelled data. Since the examples provided to the classifier have no attributed class, there is no error or reward system to confirm potential solutions. This is the main difference between unsupervised learning and supervised learning. Unsupervised learning is related to the statistical density estimation problems. However unsupervised learning also encompasses many other techniques that seek to summarize and extract the key features of the data [2][34].

2.2.1.3 Eager and Lazy Learning

In Eager Learning the aim is to fit the training data to some sort of model and generate an hypothesis of how the training data relates to the value we are trying to predict, this way spending more time doing the calculations during the training phase than the prediction phase. Unlike Eager Learning, its counterpart, Lazy Learning simply stores the training data and postpones the models computation until a test instance is requested. This makes it spend less time on the training phase but more time on the prediction phase [2][35].

2.2.2 Classifier Comparison Metrics

In this section we aim to explain and demonstrate the assessment metrics we intend to use in this study

Example Imagine there are 100 hectares of woodland, and our classifier intends to be classify the area as either burned or not. The correct predictions correspond to 80 burned hectares and 20 not burned.

2.2.2.1 Accuracy

Accuracy is one of the simplest metrics to use, where one simply compares the right guesses against the total amount of guesses made.

$$Accuracy = \frac{\text{correct guesses}}{\text{total of guesses}}$$

in the case of the example presented above it would be:

$$\frac{70 + 15}{100} = 0.85$$

2.2.2.2 Confusion Matrix

Although the overall accuracy is easy to calculate and understand it gives the map producer and user limited information. So we arrange the classifications into the table represented bellow.

Predicted class	Reference class		Total	User accuracy
	Burned	Not burned		
Burned	70	5	75	0.93
Not burned	10	15	25	0.4
Total	80	20	100	
Producer accuracy	0.875	0.25		

On the diagonal in bold we have the elements that represent the areas that were correctly classified. By defining the positive label to being the Burned label, we can say there are 70 **True Positive (TP)** and 15 **True Negative (TN)**, these labels are the elements which are correctly classified according to the ground truth. Then we have 5 **False Positive (FP)**, areas that are marked as burned but are not in fact, and 10 **False Negative (FN)**, areas that should be marked as burned but were classified as unburned.

Producer's Accuracy

The Producer's Accuracy is the map accuracy from the point of view of the map maker (the

producer). This describes how often the real features on the ground truth are correctly shown on the classified map or the probability that a certain land cover of an area on the ground is classified as such.

User's Accuracy

The User's Accuracy is the accuracy from the point of view of a map user, not the map maker. This accuracy essentially tells us how often the class on the classified map will actually be present on the ground.

2.2.2.3 Kappa Coefficient

The Kappa Coefficient measures the agreement between two or more observers including a statistic that takes into account the fact that observers will sometimes agree or disagree simply by chance [52]. The statistic formula for this purpose is:

$$k = \frac{p_o - p_e}{1 - p_e}$$

Where p_o refers to the observed accuracy, the one the classifier achieves through its predictions. p_e defines the expected accuracy, the accuracy it is expected to be achieved from the classifier, by this we mean the te combined percentage the prediction get right with the percentage of data, from each class, there is to be classified.

First we calculate the Observed Accuracy:

$$p_o = \frac{70 + 15}{100} = 0.85$$

Having the classifier predicted 70 hectares as being burned when the truth is 80 hectares, we calculate the probability of the classifier correctly predicting the burned area as:

$$p_{cb} = \frac{80}{100} * \frac{70}{100} = 0.8 * 0.7 = 0.56$$

Then we do the same for the area that is not burned:

$$p_{cnb} = \frac{20}{100} * \frac{15}{100} = 0.2 * 0.15 = 0.03$$

Having done this we calculate the Expected Accuracy:

$$p_e = p_{cb} + p_{cnb} = 0.56 + 0.03 = 0.59$$

After calculating all the elements we assemble the main formula:

$$k = \frac{0.85 - 0.59}{1 - 0.59} = \frac{0.26}{0.61} \approx 0.42$$

2.2.3 Classifiers

A classifier is an algorithm which implements a classification method. This classification method is used to infer a classification model that is able to automatically and autonomously map a label, also known as class or category, to a sample from the input data.

2.2.3.1 Artificial Neural Networks

MLP, are mathematical models that try to simulate the human brains' capacity to learn from examples and to generalize the knowledge attained from that learning to new examples. **ANN** are applicable to any problems which are rich in data but poor in models, in other words, when a solution is clearly present with a large volume of examples, that can be used as a learning base but there is no traditional method to solve it.

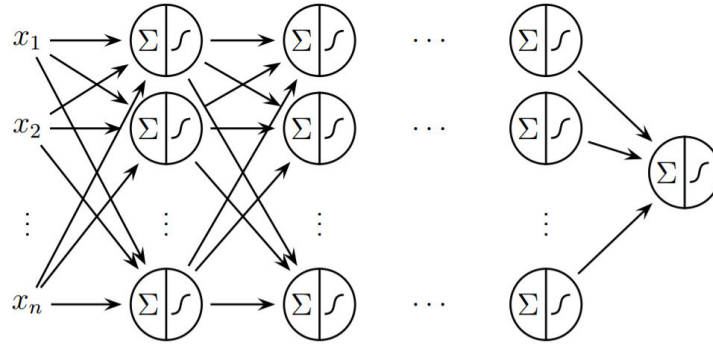


Figure 2.1: Example of a **MLP** with n inputs and one output class

source: Riedmiller [38]

The learning process consists on the iterative adjustment of the synaptic weights of the network in order to accomplish a specified task. This process is called supervised when the database is labelled and unsupervised otherwise. Each neuron that participates in the tasks receives inputs through its weighted connections. These connections represent the dendrites of the neuron. When the sum of all the signals that reach a neuron exceed a determined bias, the neuron activates and relays a output signal through its axon. The activation function of neurons are expressed by mathematical functions that provide different responses throughout their range, examples of these functions are the sigmoid, hyperbolic tangent and rectified linear function. The simplest way to group neurons in layers and form an architecture is to group them in layers and connect each neuron of each layer to each neuron of the following layer. An **ANN** that is implemented following this architectural pattern are called **MLP** and abides by some rules. Neurons of the same layer can not be connected between themselves; Signals flow only from the input layer to the output layer (forward-feeding); The number of neurons in the input layer is equal to the size of the dataset; The number of neurons in the output layer is equal to the number of classes present in the dataset; Finally, the number of hidden layers are determined for

each particular application.

Hidden Layers consist on all the layers of neurons between the input layer and the output layer.[40]

2.2.3.2 K-Nearest Neighbours

The **KNN**, is a good example of Lazy Learning. It's a simple algorithm with no training model. **KNN** basically assigns a label y to sample x based on the number of nearest neighbours k . By using a distance metric, for example Euclidean or Manhattan distance, to assert the k nearest neighbours in the training set, and by using majority voting, sample x is assigned the label y according to which label is more present amongst the k neighbours. Usually the value k is an odd number in order to avoid ties. The most common distance metric used is Euclidean, but there exist others.

Due to its simplicity, **KNN** is as widely used classification method and is quite successful in a variety of applications. The **KNN** algorithm has a large need for memory and computation, but it is very easy to parallelize on a **Single Instruction Multiple Data (SIMD)** machine [2][34][23].

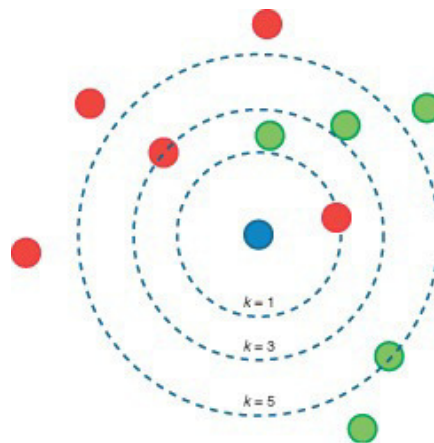


Figure 2.2: K-Nearest Neighbours with k ranging from 1 to 5

source:

<http://en.proft.me/2017/01/22/classification-using-k-nearest-neighbors-r/>

2.2.3.3 Support Vector Machines

SVM, are linear binary classifiers, in other words, they assign a given test sample a class from one of the two possible labels. The distinction between the two classes is defined by the separating hyperplane. As in opposition to a line (two dimensions) or a plane (three dimensions), the hyperplane is the definition given to every surface that has more than three dimensions.

An important consideration to have in terms of **SVM** training, is that not every available example is used in the description and specification of the hyperplane. A subset of points from the training data lie on the margin, the support vectors, these points are the only ones that define the hyperplane of the maximum margin, as seen in Figure 2.3.

The implementation of a linear SVM takes on the assumption that the feature data is linearly separable in the input space. In practice, different clusters of data point overlap, making linear separability difficult as the basic margin boundaries are insufficient to classify the patterns with high accuracy. There are different techniques than aim to tackle this problem.

In remote sensing, it is often common to identify multiple classes, for this purpose, adjustments have to be made to this simple classifier in order for it to operate in a multi-class domain. The use methods such as one-against-all, where one class is separated from all other classes, or one-against-one, which consists on the creation of all possible pairs classes from the training and the consequent separation of classes between these pairs.

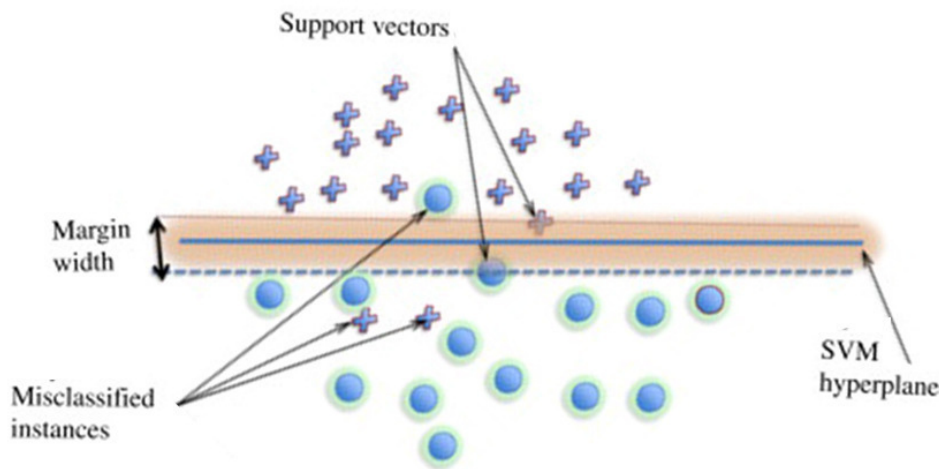


Figure 2.3: Support Vector Machine
source: Mountrakis et al. [28]

In the topic that concerns the applicability of SVMs is the choice of kernels, since they often have a weight on the results.

Kernels enable the spread of features within a new (the feature space) in a way that enables a faster and possibly better fitment of a hyperplane between them in order to separate two categories, in opposition to using a high dimensional input space for the same purpose but with increased difficulty.

There are some kernels that are not optimal for remote sensing applications producing different results [28].

2.2.3.4 Gradient Boosted Trees

Unlike decision trees, boosted trees contain a continuous score on each of the leaves. For a given example, the decision rules in the trees are used to classify it into the leaves and calculate the final prediction by summing up the score in the corresponding leaves. To learn the set of functions used in the model, a regularized objective is minimized. In

this objective function the difference between the prediction and the label is accounted in addition to the model's complexity. There also is an additional regularization term helps to smooth the final learnt weights, in order to avoid overfitting. Consequently, the regularized objective tends to select a model with simple predictive functions. The objective functions of the instance that most improves the model is greedily added to the loss function. This function is the one that defines if the estimator we are pondering to aggregate to our model brings any actual benefits in performing correct classifications.

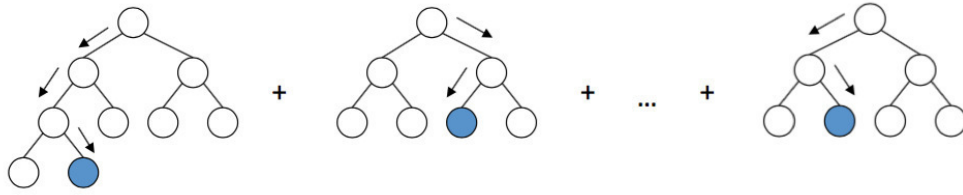


Figure 2.4: Gradient Boosted Trees
source: <http://arogozhnikov.github.io>

2.3 Parallel Processing

CPUs are optimized for low-latency access to cached datasets and control logic for unordered and speculative execution. Whereas **GPUs** are more suited for data-parallel and throughput computations with a latency-tolerant architecture with more transistors dedicated to computation [47]. This brings us to the drastic increase of the transistor density in recent years, which equates to a larger amount of thousands of millions of floating point operations per second (GFLOPS) on **GPUs** in comparison to **CPUs**. Yet, these accelerated implementations only become feasible if the portion of the algorithm intended to be parallelized corresponds to a computation-intensive and time-costly portion of the code. This is one of the problems approached by this study.

NVIDIA's GPUs are highly parallel devices which have thousands of threads running concurrently at a given time on their cores. Thus, because of their immense computational power they are much faster than **CPUs**.

CUDA Compute Unified Device Architecture, is a parallel programming paradigm released in 2007 by NVIDIA. It is used in the development of software and a variety of applications for GPUs that are highly parallel in their nature and run on hundreds of **GPU** cores. CUDA has some specific functions, called kernels. A kernel can be a function or a full program invoked by the **CPU's**. It is executed N number of times in parallel on **GPU** by using N number of threads. CUDA also has built-in shared memory and

synchronization among threads. CUDA is supported only on NVIDIA's GPUs starting from the Tesla architecture[17].

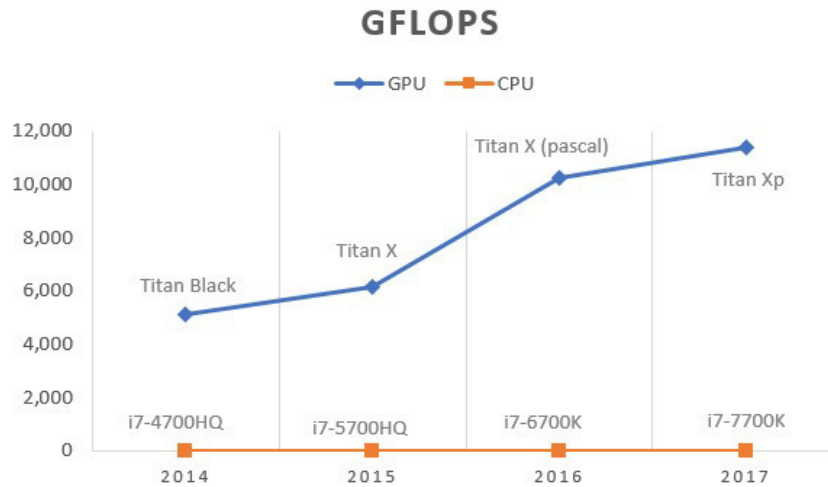


Figure 2.5: CPU vs GPU GFLOPS comparison over time

source: <https://videocardz.com/nvidia> and

https://asteroidsathome.net/boinc/cpu_list.php

2.3.1 CPU + GPU Processing

Due to the architectural differences between CPU and GPU, they excel in different tasks. Ghorpade et al. [14] compared in his article the different strengths of each hardware component. CPUs possess really fast caches, and is able to implement a fine branching granularity in addition to being able to manage a diverse assortment of processes and threads. The CPUs display a high single thread execution performance which is great for task parallelism equating to elevated performance results when executing sequential codes. On the other hand, GPUs are composed of many mathematical units with fast access to onboard memory. GPU programs run in fragments called kernels which provide high throughput in tasks with data parallelism, specially when it is of an arithmetic nature.

By executing the sequential parts of the program on the CPU and using the GPU to accelerate the data intensive part by parallelizing the data-intensive portions on many cores, the program will execute faster due to the use of the CPU on more critical tasks.

2.3.2 Python + CUDA

The major factor for the choice of a high-level, dynamic language instead of a potentially better-performing low-level static one is the complementarity of the GPU and the CPU's. The GPU is optimized to execute throughput-oriented parts of programs. This frees the CPU's to be only responsible for control and communication. This enables Python to perform this job equally well or even better than a low-level language, simply because the performance demands are reduced. As an added benefit, a high-level Python-based

compute code requires much less effort from the programmer than a low-level C-based GPU compute code. This reduction in effort derives from the data types, resources and abstractions a high-level programming language brings. Entities such as code modules and compute devices are reflected in Python using object-oriented terms this provides a better abstraction than in a low-level C interface. In Python the errors are detected and reported automatically enabling feedback to be given to the programmers [22].

2.3.3 Python Standard Library

The standard library for machine learning applications in Python is Scikit-learn, sometimes referred to as Sklearn for short. This library incorporates numerous algorithms that serve a wide range of applications, namely Classification and Clustering amongst others. The classifiers chosen from this library were:

- Multilayer Perceptron Classifier;
- Gradient Boosting Classifier;
- Support Vector Classifier;
- K Nearest Neighbour Classifier.

2.3.4 Python GPU Accelerated Machine Learning Libraries

In this section we intend to provide insights on the libraries supported in Python.

2.3.4.1 Library search and criteria

In order to find the best candidate to serve as the standard classifier counterpart numerous libraries were explored. The criteria used for the library choice was:

- Community suggestion;
- Library support from the developers;
- Library documentation and ease of use.

The libraries taken into account were Tensorflow and Keras for Artificial Neural Networks; CudaTree and XGBoost for ensembles; liquidSVM and pyKMLib for Support Vector Machines; finally knn_cuda and community implementations in Tensorflow for K Nearest Neighbours. Between Keras as Tensorflow, both posed viable choices having the possibility for low level user implementations of the algorithm, but Tensorflow proved the most user-friendly option due to having a "canned"estimators, these are already implemented and provide a plug-and-play experience in addition to its frequently updated documentation and active community. In the matter of ensembles the CUDATree estimator looked like an interesting option since it displayed an interface compatible

with the Scikit-learn library and the author claimed his implementation to have good performance, the downside was that the library had been abandoned by the author. In the meanwhile, XGBoost’s authors propose an actively developed library that also is compatible with the Scikit-learn library. For Support Vector Machines, all but the liquidSVM were either a proof of concept only mocked up as a demo or were implemented with not enough abstraction to provide ease of use to the user. liquidSVM provided a simple implementation that had an interface that was intuitive to use and was accompanied by helpful documentation and examples. The K Nearest Neighbours options all had really low-level implementations with little to none flexibility or ease of use, associated with low performance on simple tasks with small datasets, none of them proved to be an option worth seeking. In summary, we define the following libraries and their counterparts:

Table 2.5: Classifier counterpart setup

CPU-bound	GPU-accelerated
MLP Classifier	DNN Classifier
GB Classifier	XGB Classifier
SV Classifier	liquidSVM Classifier
KNN Classifier	-

The classifiers chosen were based on the ones recommended in the literature, and then further refined the options to those that had functional libraries.

In this next part we give a little insight into the each of the selected GPU libraries.

2.3.4.2 Tensorflow

TensorFlow is an open source software library for numerical computation using data flow graphs. The flexibility of the architecture allows deployment in one or more CPUs or GPUs, server or mobile using a single API. Tensorflow was originally developed by researchers and engineers working on the Google Brain Team within Google’s Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research.

2.3.4.3 XGBoost

XGBoost is short for “Extreme Gradient Boosting”, where the term “Gradient Boosting” is proposed in the paper Greedy Function Approximation: A Gradient Boosting Machine, by Friedman. It has a plugin that adds the option for GPU accelerated tree construction and prediction algorithms.

2.3.4.4 liquidSVM

liquidSVM is an implementation of [SVMs](#) whose key features are: fully integrated hyperparameter selection, extreme speed on both small and large data sets, full flexibility for experts, and inclusion of a variety of different learning scenarios.

2.3.5 Libraries used

The following libraries were the core of our experiment:

gdal 2.2.3	tensorflow-gpu 1.4
liquidSVM 1.0.1	xgboost 0.6
numpy 1.13.1	CUDA Toolkit 8.0
python 3.6.2	cuDNN v6.1
scikit-learn 0.19.0	

2.3.6 Benchmarking

The benchmarking process consists on the acquisition of the time a classifier takes to fit the data, classify the testing samples as well as the total time of all this process. Then graph are generated in order to assert trends and patterns.

2.3.7 Conclusion

The process of parallelizing machine learning algorithms in GPUs is tempting due to the increase in throughput or the decrease in the overall runtime of the program, but with it some issues must be addressed in order for the parallel version to be on par or above the CPU-bound version. The task of identification of which parts of the algorithm have the potential for parallel execution and which are confined to sequential execution, these tasks are non-trivial and have a major impact on the algorithms performance. The process of creating a parallel GPU version of state-of-the-art algorithms requires one to address the particularities of the GPU Architecture.

Python is used for the libraries mentioned in Section [2.3.4](#) due to its capability of operating at higher abstraction level than the low-level [Compute Unified Device Architecture \(CUDA\)](#) code. This enables the delegation of computation intensive tasks to GPU, while Python assumes a scripting role where it is only responsible for the overall coordination of the program.

FIRE INDICES AND FIRE PREDICTION TECHNIQUES

The aim of this chapter is to present the standard for fire risk prediction in Portugal as well related literature that highlights the diverse techniques used in programmatic approaches to this topic, creating a simplified view of the state of the art in the area.

3.1 Spectral Signature

Different surface types such as water, bare ground and vegetation reflect radiation differently in various channels. The relationship between the wavelength of electromagnetic radiation and the reflectance of a surface is defined as Spectral Signature [1].

3.1.1 Burned Area Reflectance Classification

A [Burned Area Reflectance Classification \(BARC\)](#) is a satellite-derived data layer of post-fire vegetation condition. The [BARC](#) has four classes: high, moderate, low, and unburned. [BARC](#) data is made by comparing satellite near and mid infrared reflectance values. The logic behind the process is illustrated in Figure 3.1.

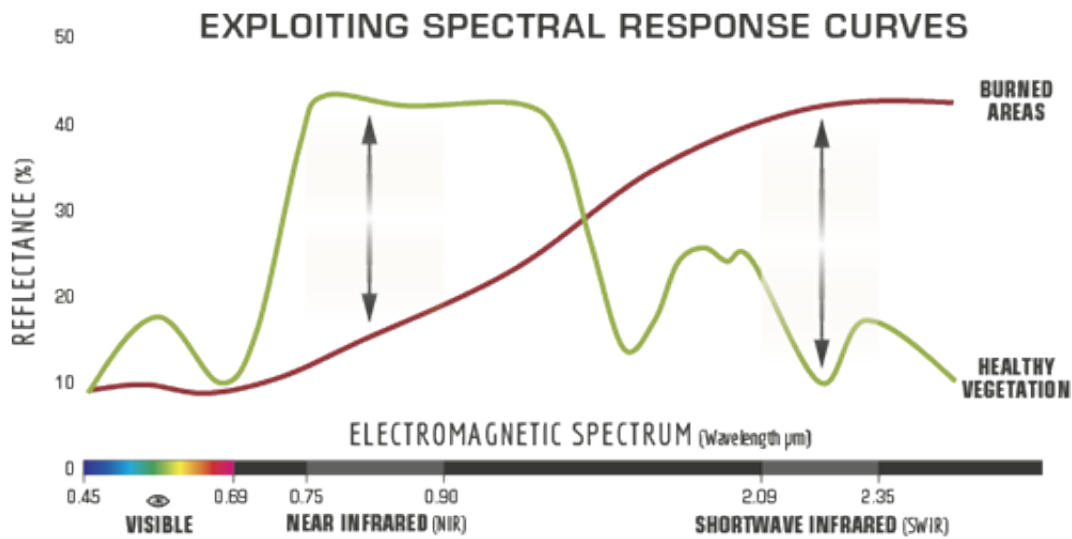


Figure 3.1: Spectral response for burned areas (BAER [5])

The Spectral Response is characterized by:

- Near Infrared - the majority of this light range is reflected by healthy green vegetation. With this one can translate to high infrared bands in areas with That means that near infrared bands will be very high in areas of healthy green vegetation and low in areas where there is little vegetation.
- Mid infrared - the majority of light is reflected by rock and bare soil. This means that mid infrared band values will be very high in bare, rocky areas with little vegetation and low in areas of healthy green vegetation.

The difference between pre and post-fire imagery displays a significative difference in its near and mid infrared values, in other words, the near values are very high and the mid values are very low in pre-fire, the opposite is verified in a post-fire scenario.

It is the relationship between these two bands that the [BARC](#) intends to exploit. the best way to achieve this goal is to measure their relationships prior and post fire. The areas with the largest differences are the best candidates for being severely burnt. The remaining areas are likely to be unburned or lightly burned. The calculation of the difference ratio is a classification of the burned areas [5].

3.1.2 Normalized Difference Vegetation Index

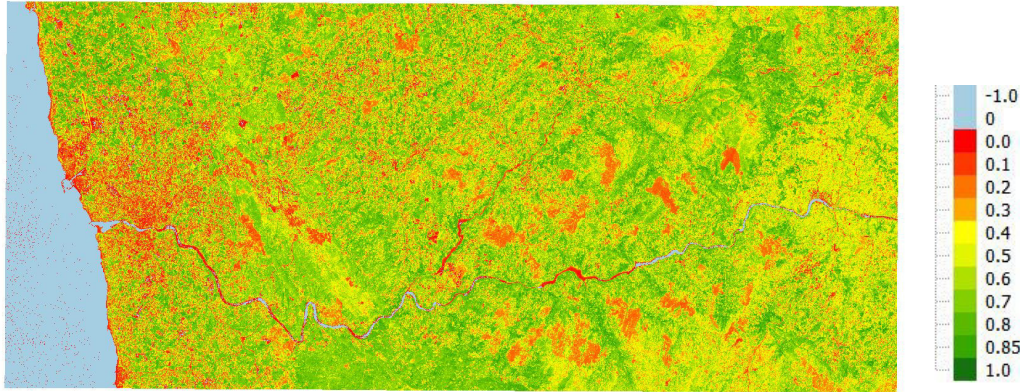


Figure 3.2: NDVI index and scale for 27th of September using Sentinel 2 imagery.

The **Normalized Difference Vegetation Index (NDVI)** is calculated from the visible and near-infrared light reflected by vegetation. Healthy vegetation absorbs most of the visible light, and reflects a large portion of the near-infrared light. Unhealthy or sparse vegetation inverts the proportions, this also occurs in burned areas due to lack of healthy vegetation cover [32]. **NDVI** is extremely sensitive to soil optical properties, and becomes difficult to interpret with low vegetation cover when the soil type is unknown [39, 45].

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

3.1.3 Soil Adjusted Vegetation Index

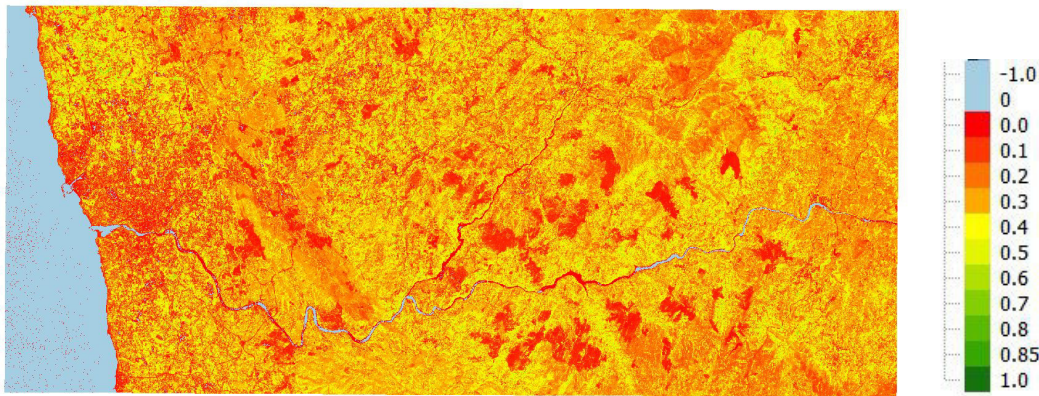


Figure 3.3: SAVI index and scale for 27th of September using Sentinel 2 imagery.

The **Soil Adjusted Vegetation Index (SAVI)** similarly adds a soil correcting factor to the formula of the **NDVI** to account for soil background variation. The calibration constant L is set to 0.5 as this value is well suited for a wide range of background brightness values and vegetation densities [39, 45].

$$SAVI = (1 - L) * \left(\frac{NIR - Red}{NIR + Red + L} \right) \text{ with } L=0.5$$

3.1.4 Moisture Stress Index

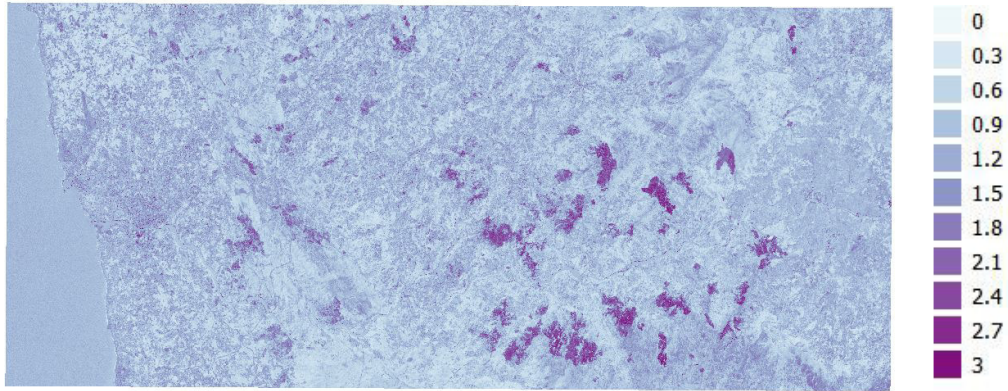


Figure 3.4: MSI index and scale for 27th of September using Sentinel 2 imagery.

The **Moisture Stress Index (MSI)** is used in the detection of leaf water content. Interpretation of the **MSI** is inverted relative to other water vegetation indices, thus higher index values indicate greater plant water stress levels and less soil moisture content can be inferred also. The values of this index range from 0 to more than 3 with the common range for green vegetation being from 0.2 to 2. **MSI** is calculated using the near-infrared and the mid-infrared reflectance [53].

$$MSI = \frac{SSWIR}{NIR}$$

3.1.5 Mid-infrared Burn Index

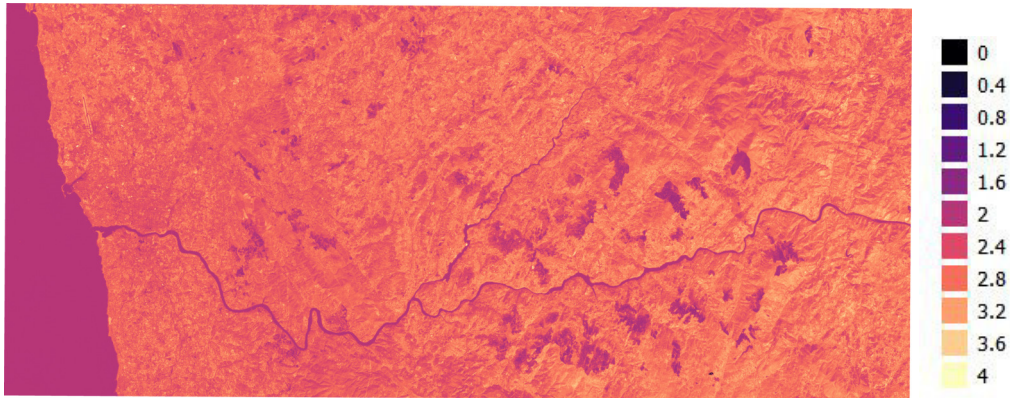


Figure 3.5: MIRBI index and scale for 27th of September using Sentinel 2 imagery.

The **Mid-infrared Burn Index (MIRBI)** was designed for a shrub/savannah vegetation type, where **NIR** wavelengths are less useful due to the deteriorated state of the vegetation during the fire season [45]. Plotting the data in **SSWIR-LSWIR** space, reveals that the vegetation (shrub and grass) and burn clusters are near-linear, near-parallel and distinctly separate. Thus, a perpendicular line should suffice to categorize the vegetation state [48].

$$MIRBI = 10 * SSWIR - 9.8 * LSWIR + 2$$

3.1.6 Normalized Burn Ratio

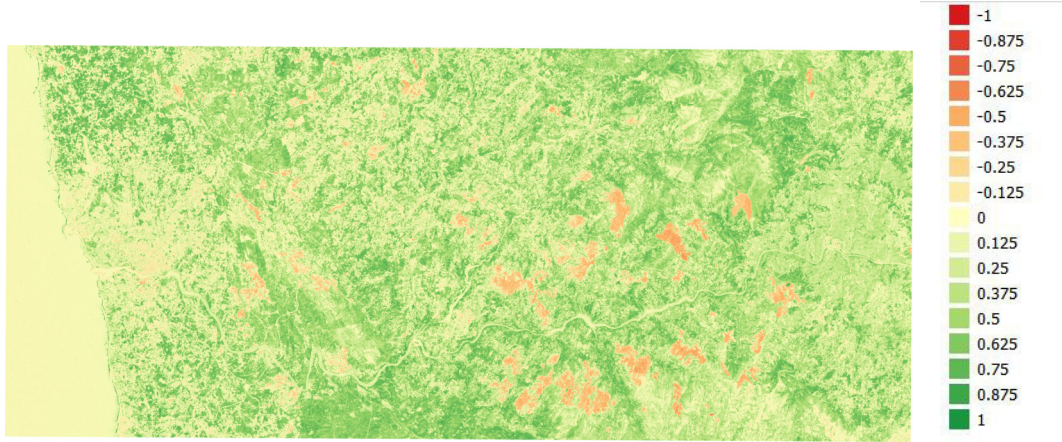


Figure 3.6: NBR index and scale for 27th of September using Sentinel 2 imagery.

The **Normalized Burn Index (NBR)** uses the **NIR** and **LSWIR** spectral region instead of the Red region as used by the **NDVI**. In the **LSWIR** region radiation is strongly absorbed by the water content in vegetation or soils. Scorching, drying, or dry soil exposure after fire will increase the **LSWIR** reflection and thereby decrease the **NBR** value. Atmospheric normalization for this index is achieved by using **NIR** and **LSWIR** [45, 49].

$$NBR = \frac{NIR - LSWIR}{NIR + LSWIR}$$

We also intend to use a few variations of the **NBR** index as well. The **Burn Ratio (BR)** corresponds to a non-normalized version of the standard **NBR**, while the remaining variations play with different combinations and ratios of the **NIR**, **SSWIR** and **LSWIR** regions.

$$BR = \frac{NIR}{LSWIR}$$

$$NBR2 = \frac{2 * NIR}{SSWIR + LSWIR}$$

$$NBR3 = \frac{SSWIR - LSWIR}{SSWIR + LSWIR}$$

For the **NBR4** index we propose a weighted contribution of the Green region of the spectrum to boost the normalizing portion.

$$NBR4 = \frac{NIR - LSWIR}{((NIR + LSWIR) * Green) + 1}$$

3.1.7 Deltas

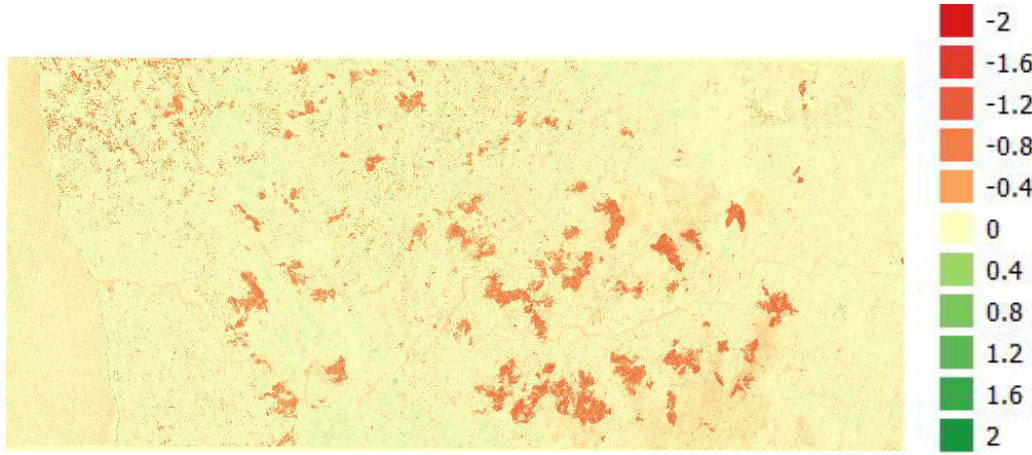


Figure 3.7: preview of the dNBR indice and scale for the interval between the 29th of July and 27th of September using Sentinel 2 imagery.

The differenced (or delta) of an index aims to isolate the burned areas from the unburned by using the difference between its pre and postfire scenarios [49]. The following expression demonstrates how it is calculated:

$$\Delta \text{Index} = \text{Index}_{\text{prefire}} - \text{Index}_{\text{postfire}}$$

3.1.8 Indice comparison

In order to validate the usefulness of the index we propose (NBR4) we decided to compare it against the standard in burned area identification.

In order to define a burned area classification for each index, all the pixels in each raster were iterated and corresponded to the correct label based on its value. In this comparison only indices generated from Sentinel 2 imagery was used due to its good resolution which will provide finer detail in order assist visual analysis.

The dNBR is one of the standards for burn severity classification but can be simplified in order to map burned areas, according to Lutes et al. [24] the threshold for values corresponding to burned areas ranges from -100 up to 1300 after the application of the proposed scaling by a factor of 10^3 . Since the lowest class considered as having any burn severity has a variation of ± 100 set our minimum value corresponding to a burned label as 200. For our proposed index, after histogram analysis of the generated raster, the decision was made of setting the threshold for values greater than zero as burned. By using these thresholds the classifications, confusion maps were generated (see Figure 3.8).

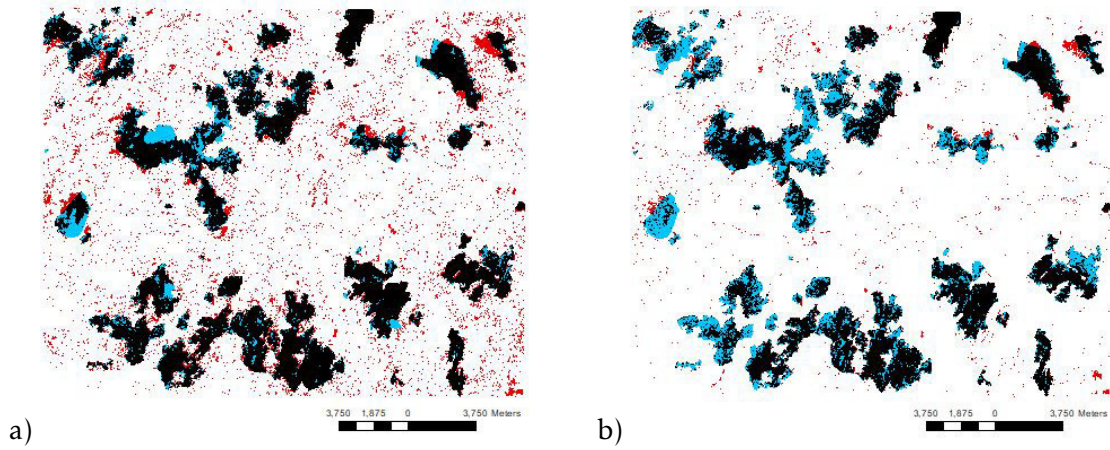


Figure 3.8: a) burned area classification confusion map using dNBR b) burned area classification confusion map using dNBR4 with Sentinel 2 imagery for September 27th

In the image generated, the blue zones correspond to areas the ground truth states are burned but the indice claims it is not. Reversely, red areas correspond to areas the indice claims are burned but are not contemplated in the ground truth. The dNBR4 index has far less false positives, but more false negatives concentrated in some regions in opposition to the dNBR index.

The use of dNBR versus RdNBR is actively debated and results regarding which metric better corresponds to field-based burn severity data have been inconclusive. While some studies have concluded that RdNBR outperforms dNBR, when using discrete classes (unchanged, low, moderate, and high) of burn severity are required, other studies have concluded that dNBR generally performs better than RdNBR, both in terms of correspondence with field measurements and overall classification accuracies [36]. The related version of the dNBR, the delta normalized burn ratio (RdNBR), is also a commonly used index for burned area identification. Based the work of Parks et al. [36], we used the variations of the dNBR described in their article that are as follow:

$$dNBR = ((NBR_{prefire} - NBR_{postfire}) * 1000) - dNBR_{offset} \quad (3.1)$$

$$dNBR = \frac{dNBR}{|NBR_{prefire}|^{0.5}} \quad (3.2)$$

In order to calculate the RdNBR, there needs to be some precautions to take care before it is calculated. Since the RdNBR reaches infinity if the pixel values in the $NBR_{prefire}$ that are equal to zero are not changed to 0.0001. Having addressed this issue, we can calculate the dNBR index and scaling it by 10^3 in order to convert the dNBR values from the float to the integer range, in resemblance to the procedure proposed by Lutes et al. [24]. Then we can proceed to the RdNBR index calculation, by using Expression 4.2. Miller et al. [27] defined the overall range of the index as going from 0 to 1300 and Parks et al. [36] also defines the threshold for burned areas at 109, leaving us with a burned area threshold of 109 to 1300. Since the lowest class for burn severity has a variation of ± 201 , we opted for

setting our burned classification threshold as starting at 310, that once analysed against the ground truth resulted in the confusion map we can see in Figure 3.9

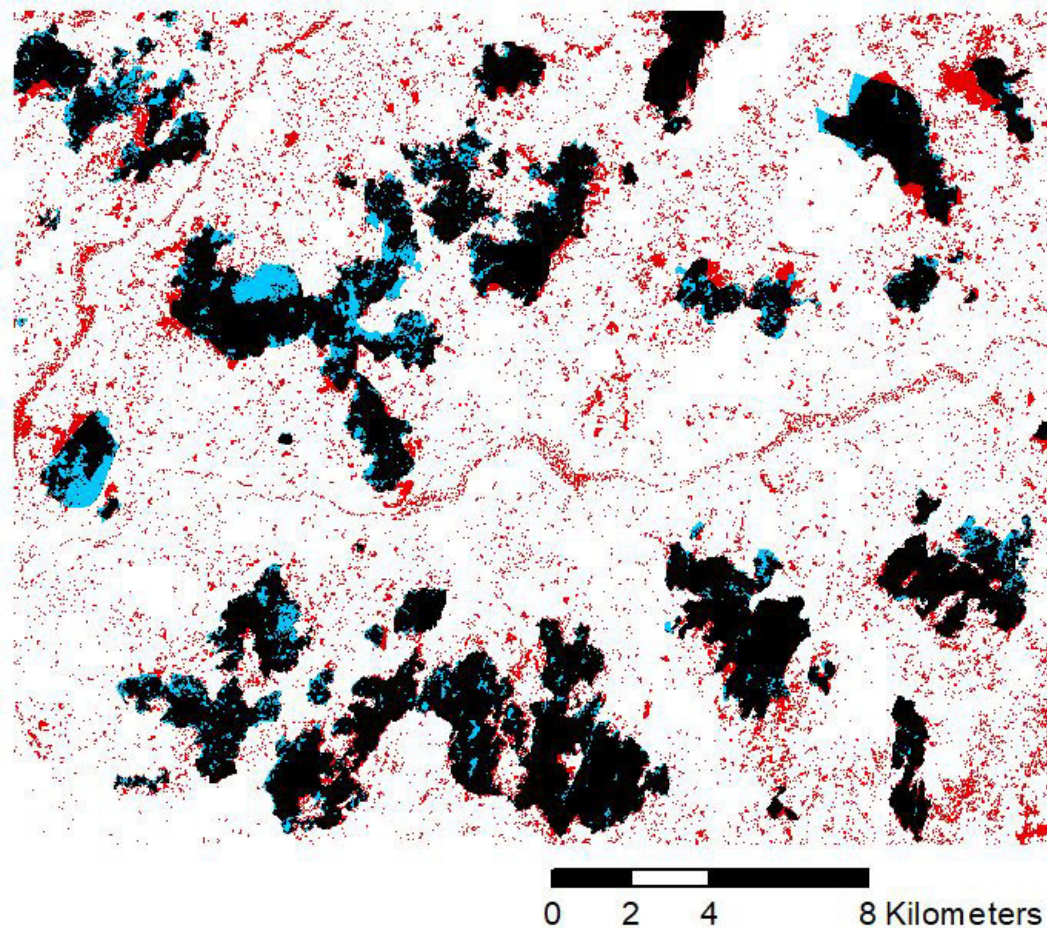


Figure 3.9: Burned area classification confusion map using the RdNBR index with Sentinel 2 imagery for September 27th

While analysing the confusion map, it is notorious the presence of the rivers appearing as classifications of burned areas in comparison to the ground truth in addition to an increase of overall false negative presence increase in comparison to the dNBR index. On the other hand, areas corresponding to false positives seem to share nearly identical footprints when comparing RdNBR with dNBR.

Resulting from this comparison study, an article was written and submitted to the VIII International Conference on Forest Fire Research. This article based itself on the conclusions drawn above and served as a proposal of the NBR4 index's use in burned area mapping.

3.2 Canadian Fire Weather Index

The Canadian Forest [Fire Weather Index \(FWI\)](#) System is comprised of six components that take into account the effects of fuel moisture and fire behaviour. Calculation of the components is based on consecutive daily observations of temperature, relative humidity, wind speed, and 24-hour rainfall. The first three components are denominated fuel moisture codes, they are used to quantify the moisture content of the diverse fuels. The remaining components are the fire behaviour indices which represent the rate of fire spread, the fuel available for combustion and the intensity of the frontal fire [33]. A diagram of the index is shown in Figure 3.10.

- **Fine Fuel Moisture Code (FFMC)** is a numeric rating of the moisture content of litter and other cured fine fuels. This code is an indicator of the relative ease of ignition and the flammability of fine fuel.
- **Duff Moisture Code (DMC)** is a numeric rating of the average moisture content of loosely compacted organic layers of moderate depth. This code gives an indication of fuel consumption in moderate duff layers and medium-size woody material.
- **Drought Code (DC)** is a numeric rating of the average moisture content of deep, compact organic layers. This code is a useful indicator of seasonal drought effects on forest fuels and the amount of smouldering in deep duff layers and large logs.
- **Initial Spread Index (ISI)** is a numeric rating of the expected rate of fire spread. It combines the effects of wind and the [FFMC](#) on rate of spread without the influence of variable fuel quantities.
- **Buildup Index (BUI)** is a numeric rating of the total amount of fuel available for combustion. It combines the [DMC](#) and the [DC](#).
- **FWI** is a numeric rating of fire intensity. It combines the [ISI](#) and the [BUI](#).

For each of these indices, the take in account the previous days results when calculating the current day's indices.

This fire index is the standard used by [IPMA](#), for measuring the risk of fire deflagration in Portugal since 2002 [18]. [IPMA](#) also generates the [Daily Severity Rating \(DSR\)](#) index based on the [FWI](#) estimation. The [DSR](#) index reflects the effort required to subdue the fire on that day [20].

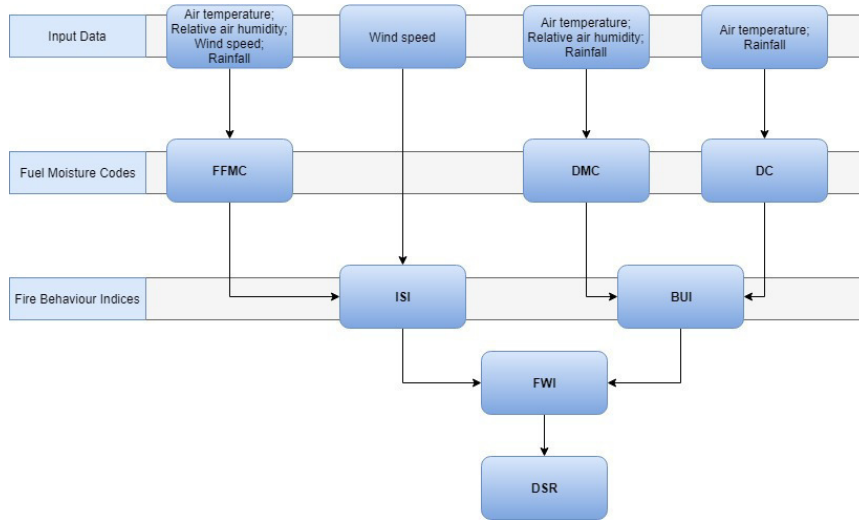


Figure 3.10: Schema of the Canadian Fire Weather Index

3.3 Fire Prediction Techniques

In this section we intend to explore what other works have been done related to the use of machine learning techniques with wildfire related tasks.

3.3.1 Burned Area Identification

Mazher [26] sets his focus on mapping a single class, burned area. The classifiers he adopted were Support Vector Machines, Support Vector Data Description and the Maximum Likelihood classifier. In order to obtain the best features a Principal Component Analysis was applied to the dataset and the Spectral Indices were calculated. The dataset is comprised of three sets of Landsat 5 imagery, derived spectral indices, original multispectral data only and a combination of both spectral indices and multispectral data. During result analysis the SVM classifier attained the best results with a kappa of 92.58%.

Barrett et al. [6] presented this study with the objective of determining the depth of burn for five large fires that occurred during the 2004, the largest fire year on record for interior Alaska. Black spruce is the dominant specie in the forested land cover of the Alaskan interior. The dataset is comprised of remote sensing data, spectral indices and other relevant information (seasonal fire regimen, meteorological data, topographic data, etc). Among the variables used there are site characteristics, which include topographic position, slope, surface statistics, Composite Burned Index (CBI), and indices generated from Landsat TM/ETM+ spectral imagery. The indices generated are NBR-family severity metrics (NBR, dNBR, RdNBR), NDVI and other spectral-band ratios. The case study provided aims to calculate the reduction in the organic layer depth. For this purpose, a decision tree was used to estimate the pre-fire depth. Field-based observations of pre-fire organic layer depth measurements were collected to serve as a baseline. To determine the organic layer depth reduction, an ensemble of regression trees was used as a base learners

and gradient boosting was applied to boost accuracy. The regression trees obtained a squared error of 0.60 and 0.55 for relative and absolute reduction of depth respectively. Through the boosting algorithm, squared error scores of 0.80 and 0.79 were obtained (with the removal of an extreme outlier in the absolute path, 0.84 was obtained, versus the previous 0.80). The maps of absolute and relative depth reduction reflect the proportion of the landscape covered by the output fire severity values.

The objective of Petropoulos et al. [37] study was to investigate the use of SVM classifiers combined with multispectral Landsat TM image for obtaining burnt area mapping in Greece. The dataset was comprised of the first four bands from a Landsat-5 TM multispectral image dated 2 days after the fire suppression. The SVM classifier performed classification using Radial Basis Function (RBF), polynomial, linear and Sigmoid kernels. The ground-truth for this study was defined by using the burnt area map from the Risk-EOS Burnt Scar Mapping service for this specific fire event. The overall classification metrics resulted in a mean accuracy of 95.87% (93% to 96% overall) and a mean kappa coefficient of 0.948 (0.920 to 0.956 overall), with the burnt area class always appearing clearly separable from all the other classes used in the classification scheme (agricultural areas, forests, scrubland/herbaceous vegetation, and urban fabric/bare soil areas).

3.3.2 Fire Risk Classification

Sakr et al. [43] compares two artificial intelligence based methods, Artificial Neural Networks (ANN) and Support Vector Machines (SVM). A binary classification and multi-class classification scenarios were used to compare the accuracy SVM and ANN. The features for the experiment were the minimal and maximal temperature, solar radiation and windspeed of the day and the cumulative precipitation level. Both achieved high accuracy predictions. Additionally, it showed that ANN outperforms SVM in multi-class classification by 0.17 fires, while SVM outperforms ANN in the binary classification of fire/no fire scenario.

Arpaci et al. [3] aimed to identify the main factors of wildfire distribution, while comparing the Maximum Entropy and Random Forests machine learning algorithms, in order to model and generate a map of potential ignition of the region of Tyrol. For this purpose, a dataset containing a wide range of topography, vegetation, climate and socio-economical features was used. The experiment resulted in a satisfying performance of both models, with either one being able to identify potential fire locations. The AUC score in testing was 0.809 for MaxEnt and 0.816 for RF.

Naganathan et al. [29] looked to understand the reliability of SVM, KNN and DT models by evaluating their performance in binary and multiclass classification of wildfire occurrence and severity. The dataset was comprised of meteorological (maximum and minimum temperature, humidity, precipitation and snowfall) and fire data (Burnt area, severity, latitude, longitude). The experiment resulted in the SVM classifier scoring the best accuracy in binary classification (65%) and KNN in the multiclass classification

(44%).

Stojanova et al. [46] developed a study with the aim of building improved models that predict the risk of fire outbreaks in Slovenia. For this purpose a wide range of classifiers was used. Among the single classifiers were included k-Nearest Neighbors, Naive Bayes, J48 decision trees, jRIP classification rules, Logistic regression, Support Vector Machines (SVM), Bayesian Networks. As for the ensemble methods the Boosting, Bagging and Random Forests of decision trees were taken into account. The data used was split into four datasets, continental Slovenia (Slovenia dataset), coastal Slovenia (Coastal dataset) and Kras region (Kras dataset) with and without Lidar. The number of attributes ranging from 106 to 159. These datasets are composed of information about the vegetation, terrain characteristics, meteorological observations, the percentage of human-made structures in the quadrant as well as the fire/no examples compiled by the authors. The results showed that in overall the RF classifiers delivered the best results throughout the assortment of datasets.

Safi and Bouroumi [41] wrote a paper about the prediction of forest fires using ANN. For this endeavour four FWI components were used (FFMC, DMC, DC and ISI, the remainder were excluded due to their dependence to the previous values) in conjunction with meteorological, temporal and spatial features. Two topologies were tested, topology A with two hidden layers (12 and 6 neurons), and topology B with a single hidden layer with 36 neurons. Topology B obtained then best error rate (5%) with 10000 iterations.

Karouni et al. [21] intended to used datamining techniques to predict fire occurrence in the Northern Lebanon region. The classifiers selected for the purpose were Neural Networks and Decision Trees using four meteorological features (temperature, relative humidity, wind speed and precipitation). Even though the performance of the Decision Tree was not stated in the paper, it enabled to understand which were the most important features contained in the dataset. As for the Neural Network, it was executed with an assortment of different sets of features to compare performances, these ranged from 91,1% to 95,6%.

Sakr et al. [42] proposed a fire prediction architecture purely based on the SVM classifier. This architecture aims to classify a specific day with an index that relates to the number of fires that may occur. The index ranges from 1 (lowest fire risk) to 4 (highest fire risk). The architecture is composed by three SVMs that on the first level classifies the data into belonging either to index 1, 2 or index 3, 4. On the second level each group is forwarded to an SVM, that finally labels the data with a single index. The dataset contains meteorological observation (min/max temperature, humidity, solar radiation, wind speed and cumulative precipitation) in addition to the number of forest fires. The presented mechanism presented a very high accuracy rating, ranging from 78,4% way up to 96%.

3.3.3 Summary

Table 3.1: Simplified Table of related works

Authors	Objective	Recommended Classifiers
Mazher [26]	Identification of Burned Areas	SVM
Barrett et al. [6]	Identification of Burned Areas	GB
Petropoulos et al. [37]	Identification of Burned Areas	SVM
Arpaci et al. [3]	Prediction of wildfires	RF, MaxEnt
Naganathan et al. [29]	Prediction of wildfires	SVM and KNN
Stojanova et al. [46]	Prediction of wildfires	RF
Safi and Bouroumi [41]	Prediction of wildfires	ANN
Karouni et al. [21]	Prediction of wildfires	ANN and DT
Sakr et al. [42]	Prediction of wildfires	SVM

3.3.4 Conclusion

As we can see, tasks related to wildfire activity as been studied in association with a wide variety of machine learning algorithms. These studies have achieved in overall promising results to efforts in the areas of both burned area identification and risk of wildfire classification with some of them obtaining over 95% overall accuracy.

EXPERIMENTAL STUDY

4.1 Methodology

The objective of this study is to evaluate the performance of several algorithms by benchmarking python's Scikit-learn library classifiers against their parallel GPU counterparts. For this task we will use images from different satellites. In particular, we will evaluate the performance of the [MLP](#), [GB](#), [SVM](#) and [KNN](#) classifiers using Sentinel-2, Landsat 8 and MODIS imagery with a burned area identification task. Our work methodology can be divided into a pre-classification phase which is followed by a classification and a complementary studies phase.

4.1.1 Pre-Classification Phase

In this phase the needed rasters are downloaded and processed. After doing this we proceed to the indice generation using the processed raster as well as a comparison of some of the generated indices.

4.1.2 Classification Phase

With all images pre-processed the classification process starts by performing a feature selection followed by parameter fine tuning (explained more explicitly in sections [4.3](#) and [4.4](#)) and finally the classification process itself. In sum, this phase can be decomposed into the following steps:

1. Perform feature selection with the full breadth of features available.
2. Perform parameter fine-tuning for each of Scikit-learn's classifiers.
3. Train both the standard and [GPU](#) classifiers with the same optimal parameters.

4. Perform testing on the final model and gather statistics on the satellite-classifier-training set combination.

4.1.3 Complementary Studies Phase

In this phase we delve into other experiments, like daily burned area classification and identification of wildfire hazard zones, using what we learned in the previous phases without studying too in depth. This provides a greater understanding of the context and paves the way for future work.

4.2 Raster processing

After downloading, the different raster were submitted to a correction process. For the Sentinel 2 and Landsat 8 rasters these underwent a [Dark Object Subtraction 1 \(DOS1\)](#) process, while [MODIS](#) rasters were reprojected to WGS 84 coordinate reference system.

4.3 Feature Selection

In order to perform a feature selection, we applied a LASSO regression to the dataset containing all Sentinel-2 pre and post-fire scenario band data and indices, as well as indice variation between the sensing dates of July 29th and September 27th (deltas). The Lasso regression was run several times with different values for the weight parameter. Choosing a single weight value and opt for the resulting feature selection would turn our choice biased. A solution found for the problem was to define an heuristic to help us choose the most relevant features, in this case an average. The mean value for each feature across all the different weight values being greater than zero equates to that particular feature having some relevance. The features selected with this procedure were the ones represented on figure 4.1. The remaining features had a mean score of zero, meaning that they are not as relevant as the ones with non-zero values. We used the Sentinel-2 imagery due to it having the best resolution of the three, which equates to a greater amount of data points to assist in finding the best combination of features for our models.

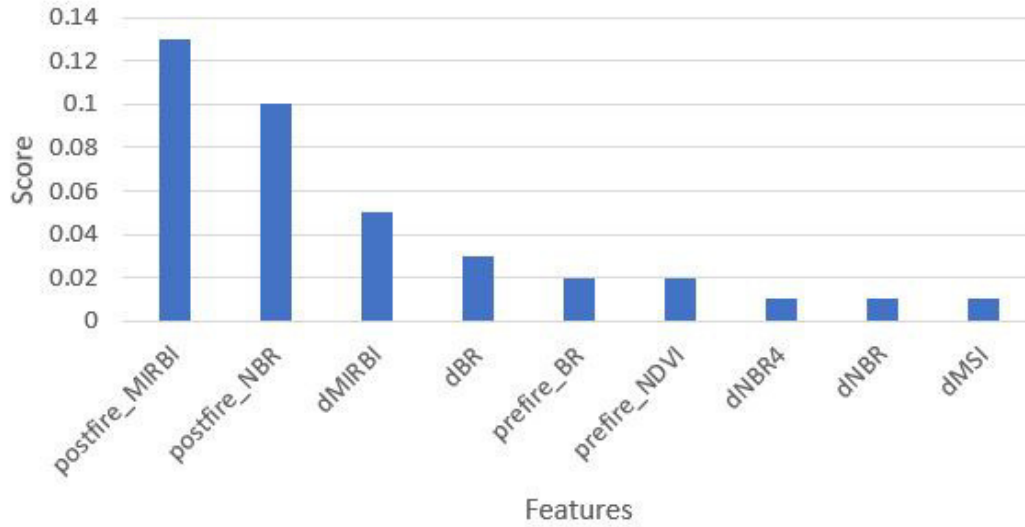


Figure 4.1: Graph of the non-zero LASSO score features

The feature selection provided some insight into which features would provide greater assistance in order to perform the task. It revealed the relevance of pre-fire features ([BR](#) and [NDVI](#)), post-fire features ([MIRBI](#) and [NBR](#)) that were considered the most important due to their high score, but the largest amount of features present corresponded to the difference features which aim to highlight the changes between the two dates.

4.4 Parameter Tuning

To discover which were the best parameters for the classifiers used in section [4.5](#) we performed a 5-fold cross-validation over several parameter values. The data used for parameter tuning was from MODIS imagery in order to account for the bottleneck in samples and resolution that is associated to it. The best parameters resulting from tuning with this data in order to level the field when it takes to classifier comparison, which is the main focus of our work. We only performed parameter tuning on the Scikit-learn classifiers in order to establish a base-line between them and their [GPU](#)-accelerated counterparts.

4.4.1 Multilayer Perceptron Classifier

For this classifier we tuned the number of neurons in each hidden layer (`hidden_layer_sizes`) from (1,1) to (9,9). The best setting for the hidden-layers was (9,9) with 89,3% accuracy.

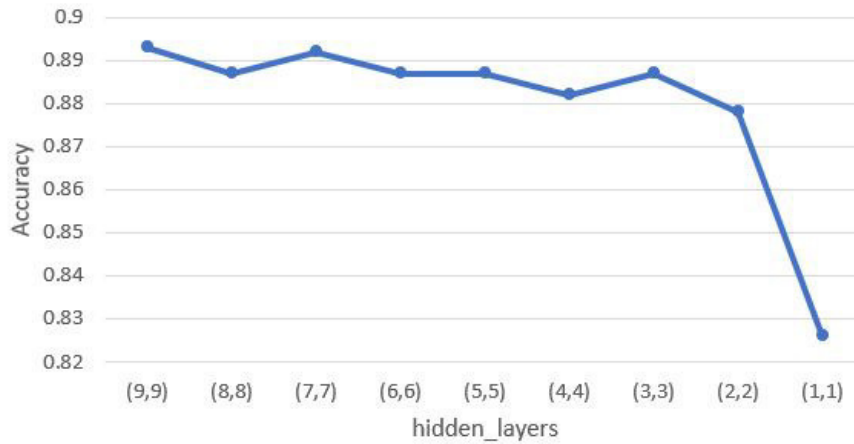


Figure 4.2: Multilayer Perceptron hidden_layer_sizes during Cross Validation

4.4.2 Gradient Boosting Classifier

The parameters chosen for tuning were the maximum depth limits the number of nodes in the tree (`max_depth`) and the minimum number of samples required to split an internal node (`min_samples_split`). These parameters ranged from 1 to 10 and 2 to 10 respectively. The best combination was `max_depth` at 4 and `min_samples_split` at 2 with an accuracy of 89,1%.

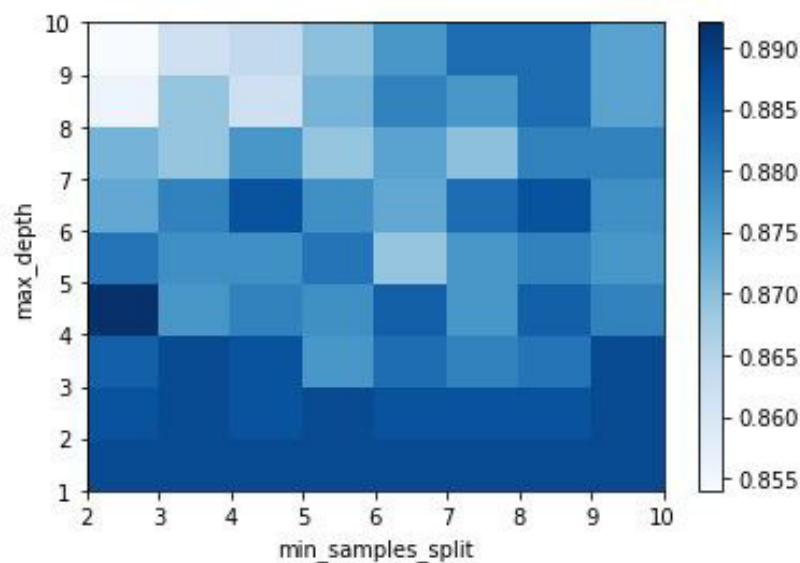


Figure 4.3: Gradient Boosting `max_depth` and `min_samples_split` during Cross Validation

4.4.3 Support Vector Classifier

The parameter chosen for tuning was the Gamma coefficient. This parameter was firstly tuned with a wider range of values. After discovering the preliminary best value for Gamma, we shortened the range around it and set a smaller step. The final best value discovered was a Gamma of 0.39 with an accuracy score of 0.89.

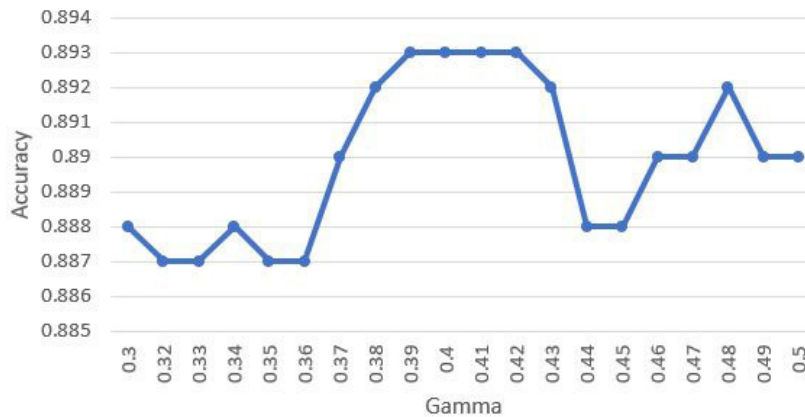


Figure 4.4: Support Vector Machines Gamma parameter during Cross Validation

4.4.4 K Nearest Neighbours Classifier

For this classifier we decided to only use odd numbers for the Nearest Neighbours parameter (K) in order to prevent ties while classifying. The best value found for K as 5 with an accuracy score of 87,1%. This process is depicted in figure 4.5.

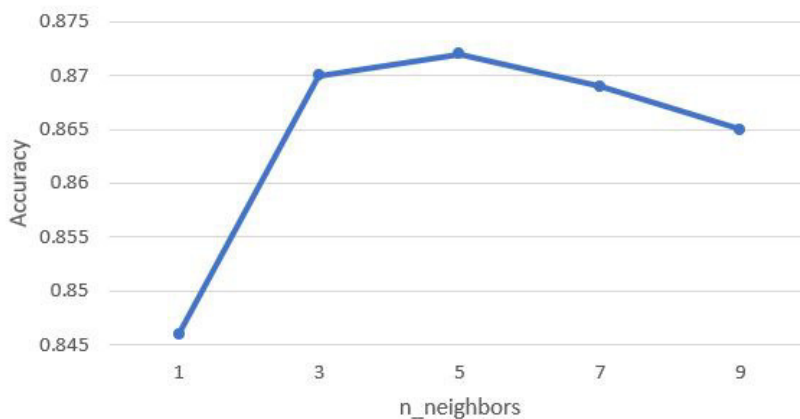


Figure 4.5: K Nearest Neighbours n_neighbors during Cross Validation

4.5 Classification Process

The classification process is staged in an incremental fashion, where the percentage of the training set used is gradually increased on each run. The training set percentages used for this experiment were 1, 5, 10, 20 and 30 percent respectively. The data used in this phase is comprised of the pixel values of satellite imagery that was subsequently split into two sets, according to the training set percentage assigned for that run. In order to maintain the ratio between both classes, we used a stratified split. The amount of samples per percentage corresponding to each dataset is specified in Table 4.1.

Table 4.1: Sample amounts per training set percentage.

Training set percentage	1	5	10	20	30
Sentinel 2	55931	279655	559311	1118622	1677934
Landsat 8	6219	31099	62198	124397	186595
MODIS	20	110	203	406	609

The classification process starts by splitting the pixel values from the previously processed rasters into a training and testing sets. The first one is used to train the classifier which is then fed the testing set. From this testing set, the classifiers generates predictions that are compared with the true values of the testing set in order to generate metrics and statistics. The trained classifier then performs a classification of all the available data that is then saved in a raster format, in order to visually assess the classification for the entire area of study. The classification process is illustrated in figure 4.6.

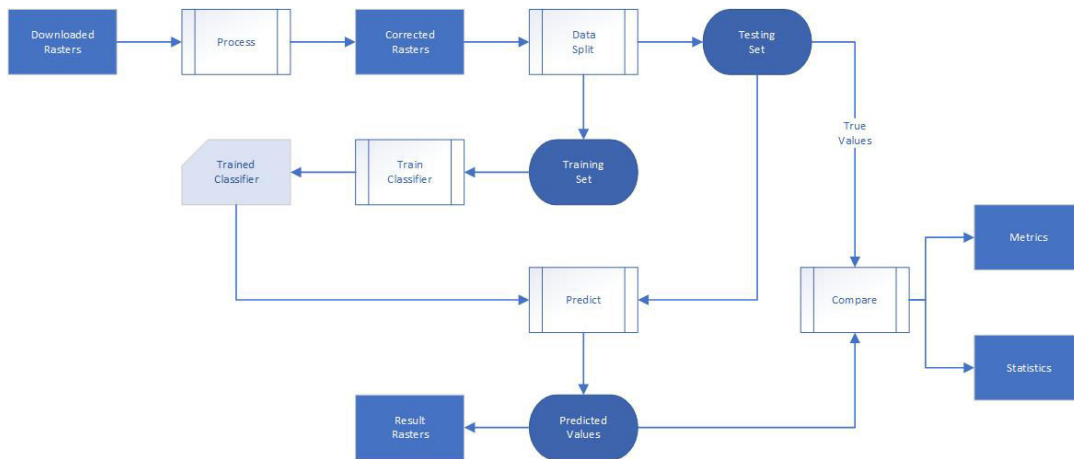


Figure 4.6: Diagram of the classification process

4.6 Experimentation setting

For this experiment we intend exhaust all the possible combinations of training set sizes, satellite imagery and classifiers, this procedure resembles a brute-force approach. This

will give us greater insight to the characteristics of each classifier while letting us observe how it reacts with data provided by each satellite.

Brute-force In computer science context, brute-force is a very general way of finding solutions, this method consists of enumerating all possible solution candidates and checking which candidate solution satisfies/better satisfies the problem. It is very easy/simple to implement, and will always find a solution if it exists. It has, however, a computational cost proportional to the number of candidate solutions. This reason, is, the obvious main deterrent of it being chosen to solve all kind of problems as most of the times, Time is a very important factor/resource to consider.

4.6.1 Hardware specifications

The machine used for this experiment had the following configuration:

- Intel Core i7 6700HQ @ 2.60 GHz
- 16GB DDR4 RAM @ 1200MHz
- NVIDIA GTX950M 4GB

4.7 Result Discussion

In this section we proceed to the analysis of the generated data and metrics.

4.7.1 Organization of result discussion

When comparing each classifier match-ups, we will follow the order of comparing the [CPU's](#)-bound classifier first by analysing its execution times and accuracy assessment metrics and performing the same steps for its [GPU](#) counterpart. After each pair has been analysed alone, a head-to-head comparison will take place. Finally, a general summary will be made highlighting key-points.

4.7.2 Artificial Neural Networks

In this section we will compare the Artificial Neural Networks algorithm using the [MLP](#) and DNN classifier implementations.

4.7.2.1 MLP

Execution Time

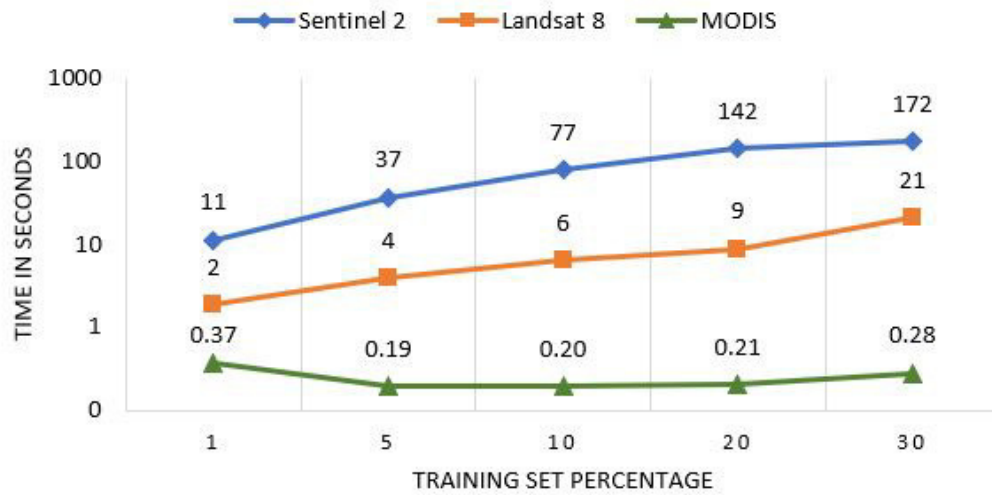


Figure 4.7: Total execution times of the MLP classifier across all satellite datasets and training set percentages

Confusion Matrix

Table 4.2: Confusion Matrix from the MLP Classifier using 10% training data from Sentinel 2 imagery

Predicted class	Reference class		Predicted Total	Producer Accuracy
	Not Burned	Burned		
Not Burned	3979586	137911	4117497	0.97
Burned	83283	833023	916306	0.91
Reference Total	4062869	970934	Overall Accuracy	0.96
User Accuracy	0.98	0.86	Overall Kappa	0.86

4.7.2.2 DNN

Execution Time

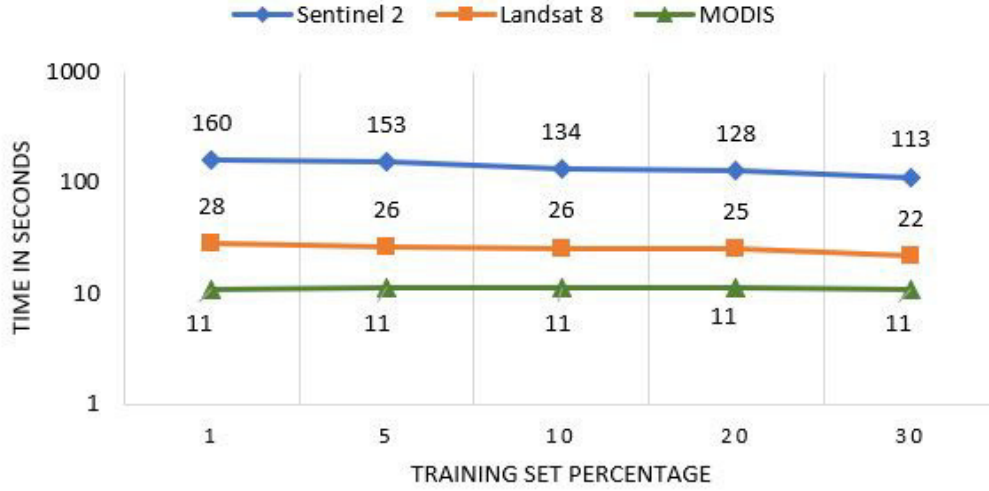


Figure 4.8: Total execution times of the DNN classifier across all satellite datasets and training set percentages

Confusion Matrix

Table 4.3: Confusion Matrix from DNN Classifier using 30% training data from Sentinel 2 imagery

Predicted class	Reference class		Predicted Total	Producer Accuracy
	Not Burned	Burned		
Not Burned	3094174	155551	3249725	0.95
Burned	65835	599620	665455	0.90
Reference Total	3160009	755171	Overall Accuracy	0.94
User Accuracy	0.98	0.79	Overall Kappa	0.82

4.7.2.3 Comparison

When comparing the execution times of the Multilayer Perceptron Classifier and the Deep Neural Network Classifier, we can see an expectable behaviour from the first, where the execution time rises as the training set percentage increases. On the other hand, the DNN classifier tends to follow the opposite trend, decreasing its execution time as the training set percentage increases. This may derive from the training time being constant independent of the amount of training samples (see Appendix A.2), this leads to a decrease in the total execution time due increasingly smaller amount of samples to test. This is visible in all the satellite data, but when using the MODIS dataset, the training time represents almost the total amount of time of the execution. The Kappa values from the DNN classifier tend to be lower than the ones from the MLP classifier, this may be justified by the Deep Learning nature of the classifier where its performance may increase

with larger amounts of training samples. Whereas the MLP may be more suitable to be used in smaller datasets. Yet, if the datasets become bigger than the Sentinel 2 dataset, the accuracy-speed tradeoff may be feasible if the execution times are a constraint.

4.7.3 Gradient Boosting

In this section we will compare the Gradient Boosting algorithm using the GB and XGB classifier implementations.

4.7.3.1 Gradient Boosting Classifier

Execution Time

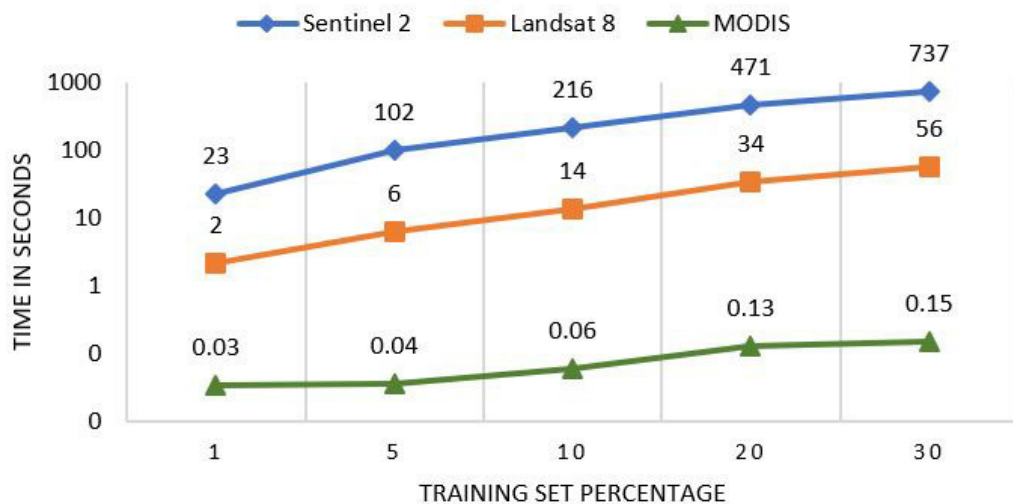


Figure 4.9: Total execution times of the GB classifier across all satellite datasets and training set percentages

Confusion Matrix

Table 4.4: Confusion Matrix from GB Classifier using 10% training data from Sentinel 2 imagery

Predicted class	Reference class		Predicted Total	Producer Accuracy
	Not Burned	Burned		
Not Burned	3974394	133914	4108308	0.97
Burned	88475	837020	925495	0.90
Reference Total	4062869	970934	Overall Accuracy	0.96
User Accuracy	0.98	0.86	Overall Kappa	0.86

4.7.3.2 XGBoost

Execution Time

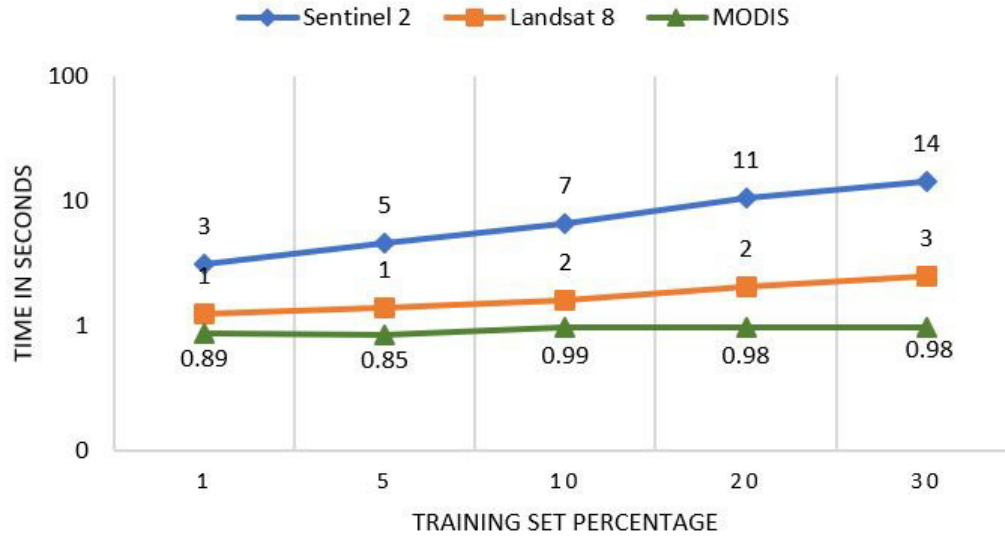


Figure 4.10: Total execution times of the XGB classifier across all satellite datasets and training set percentages

Confusion Matrix

Table 4.5: Confusion Matrix from XGB Classifier using 10% training data from Sentinel 2 imagery

Predicted class	Reference class		Predicted Total	Producer Accuracy
	Not Burned	Burned		
Not Burned	3975301	134450	4109751	0.97
Burned	87568	836484	924052	0.91
Reference Total	4062869	970934	Overall Accuracy	0.96
User Accuracy	0.98	0.86	Overall Kappa	0.86

4.7.3.3 Comparison

Both the XGBoost classifier and Gradient Boosting classifier maintain the trend of increasing with as the training set percentage also increases. In the Sentinel 2 and Landsat 8 dataset XGBoost displays the fastest times, spending the most of the time training and only a small fraction of the time classifying (see Appendix A.4). Using the MODIS dataset, the tables are turned with the GB classifier being faster. This fact may be due to the time it takes to allocate the GPU for processing.

4.7.4 Support Vector Machines

In this section we will compare the Support Vector Machines algorithm using the SVC and liquidSVM classifier implementations.

4.7.4.1 Support Vector Classifier

Execution Time

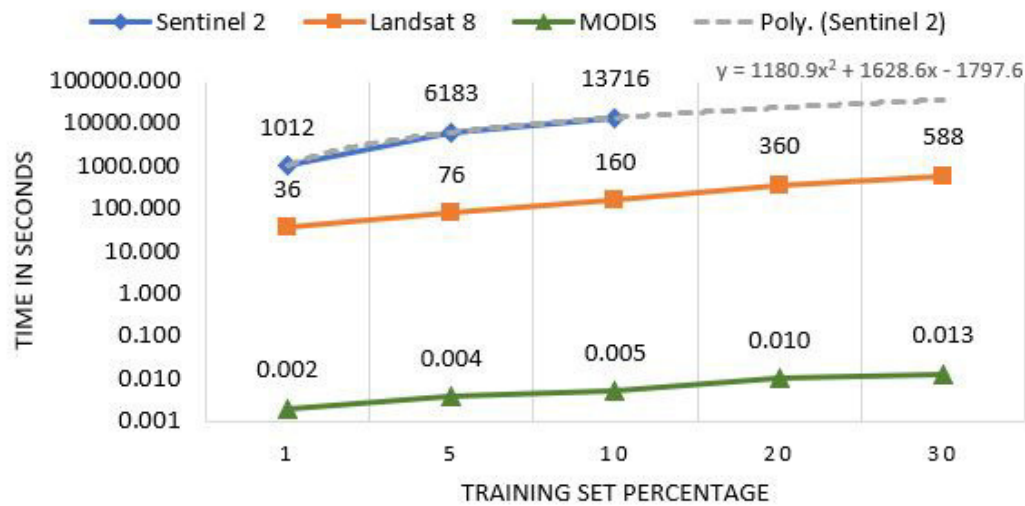


Figure 4.11: Total execution times of the SVC classifier across all satellite datasets and training set percentages

Confusion Matrix

Table 4.6: Confusion Matrix from SVC classifier using 5% training data from Sentinel 2 imagery

Predicted class	Reference class		Predicted Total	Producer Accuracy
	Not Burned	Burned		
Not Burned	4203848	162079	4365927	0.96
Burned	84737	862795	947532	0.91
Reference Total	4288585	1024874	Overall Accuracy	0.95
User Accuracy	0.98	0.84	Overall Kappa	0.85

4.7.4.2 liquidSVM Classifier

Execution Time

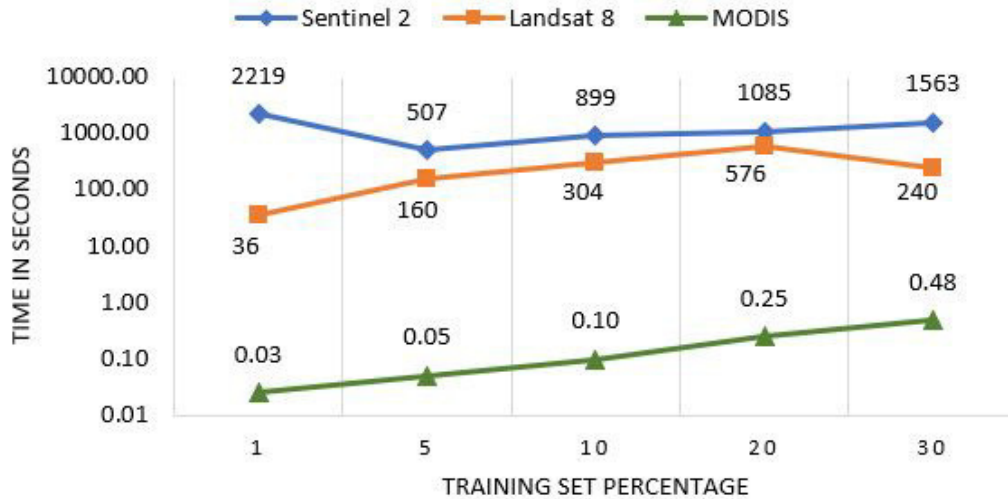


Figure 4.12: Total execution times of the liquidSVM classifier across all satellite datasets and training set percentages

Confusion Matrix

Table 4.7: Confusion Matrix from liquidSVM Classifier using 10% training data from Sentinel 2 imagery

Predicted class	Reference class		Predicted Total	Producer Accuracy
	Not Burned	Burned		
Not Burned	3974205	132740	4106945	0.97
Burned	88664	838194	926858	0.90
Reference Total	4062869	970934	Overall Accuracy	0.96
User Accuracy	0.98	0.86	Overall Kappa	0.86

4.7.4.3 Comparison

With the liquidSVM classifier the execution times are somewhat inconsistent, where with large datasets such as Sentinel 2, it tends to be lot faster that the SVC classifier, while with the Landsat 8 and MODIS dataset, it is slower and presents odd behaviours in the time it takes to execute. This may derive from the underlying algorithm implementation. For the SVC classifier it displays a typical non-linear behaviour (in the worst case) as the training set size increases. For the Sentinel 2 dataset it was only able to be tested until the 10% training set size due to time constraints, but by estimating its other values, it would roughly equate to 140 and 307 hours for the 20 and 30 percentage of training set size.

4.7.5 K Nearest Neighbours

In this section we will record the performance of the standard KNN classifier as example of how a non-parametric algorithms performs in this experimental setup but also as milestone when comparing all others.

Execution Time

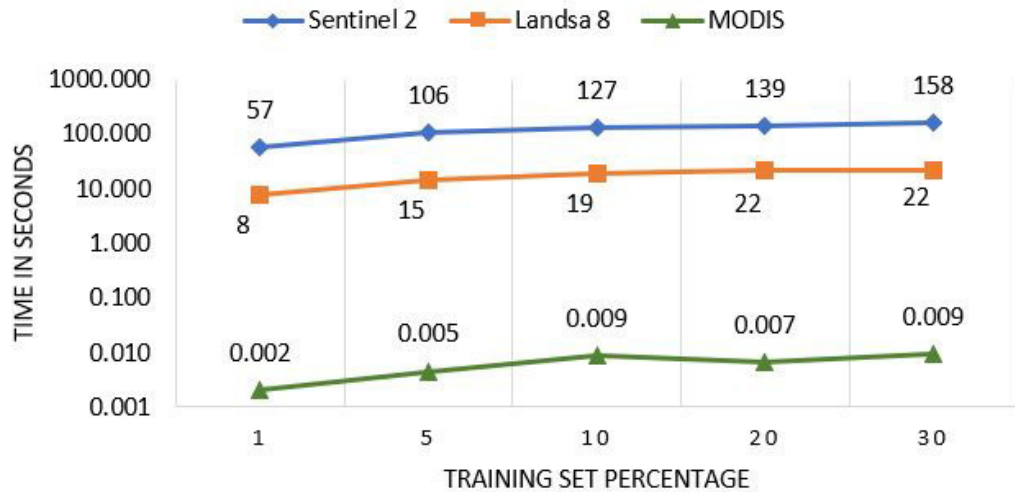


Figure 4.13: Total execution times of the KNN classifier across all satellite datasets and training set percentages

Confusion Matrix

Table 4.8: Confusion Matrix from K Nearest Neighbours Classifier using 10% training data from Sentinel 2 imagery

	Reference class			
Predicted class	Not Burned	Burned	Predicted Total	Producer Accuracy
Not Burned	3970301	144358	4114659	0.96
Burned	92568	826576	919144	0.90
Reference Total	4062869	970934	Overall Accuracy	0.96
User Accuracy	0.98	0.85	Overall Kappa	0.86

Analysis

Despite not having previously decided on a suitable counterpart to compare with we found it would be interesting to view how such a simple classifier performs with different datasets. While using the Sentinel and Landsat dataset it maintains a stable increases to the total execution time. With the MODIS dataset since there are so few samples, in spite the times fluctuating a little, the total time stays in the same range never exceeding 0.009 seconds which is perceived as an "instant" classification.

4.7.6 Comparison

Using the results over multiple runs using several combinations of classifiers, data sources and training set percentages, whisker box charts were created to study the behaviour of both the accuracy (figure 4.14) and kappa (figure 4.15) across these combinations.

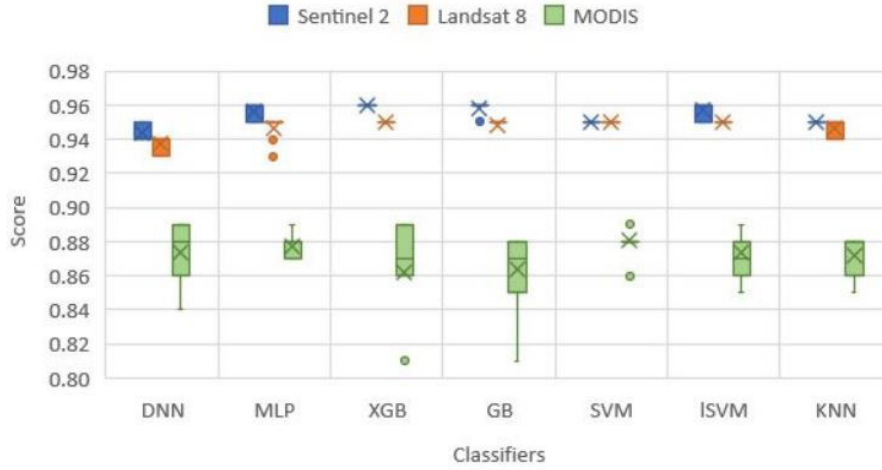


Figure 4.14: Overall accuracy.

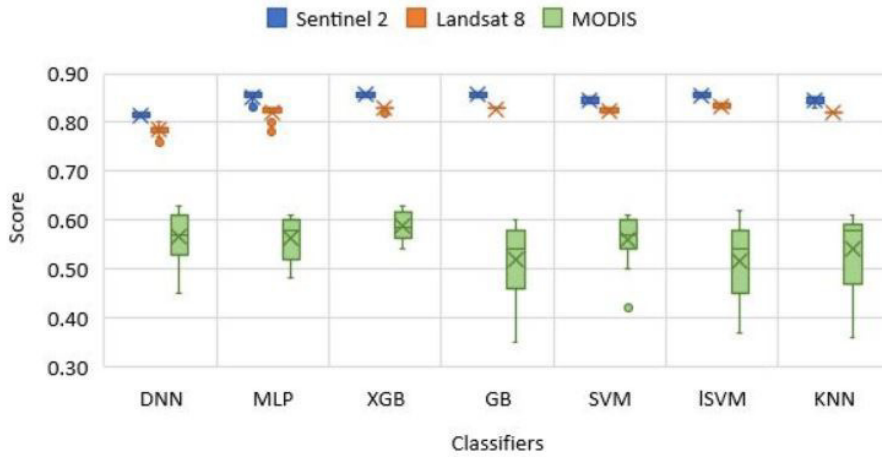


Figure 4.15: Overall kappa.

Based on the results illustrated in figures 4.14 and 4.15, as well as Appendix A, we can infer that the best models for Sentinel 2 and MODIS were generated using XGBoost classifier with 10% and 20% training set percentages respectively. For Landsat 8 the best models were generated using the liquidSVM classifier using 30% training data. Using classifications from these models confusion maps were generated. While analysing these maps, it becomes evident that they present some gaps inside the areas in comparison to their respective ground truths. This may be caused by the model opting for defining some areas in them contained as not being burned, either for being man-made structures such as quarries or buildings (which exist within the study area) or natural rock formations. These confusion maps are represented in figure 4.16. It is also important

to state that the accuracy loss when comparing Sentinel 2 with Landsat 8 classifications is that the latter appear to possess more noise in its imagery. Since the MODIS cell size derived from its resolution is so wide, a sampling criteria for defining if a pixel is burned or not is vital, in order to make sure that the ground truth remains reliable enough to correspond to the truth on the field. For this purpose, an average criteria was chosen, this criteria attributes to the "new" pixel the category of the , this assures that the category attributed is the one that has largest presence within that cell.

A general trend seen in the confusion matrices for all algorithms is tradeoff between burned and not burned values when comparing the producer and user accuracies. The only justification found for this fact is the one evidenced in figure 4.16, where the algorithms "correct" the areas in the ground truth, rooting out areas they decide that are unburned and adding areas previously left not contemplated based in their feature values.

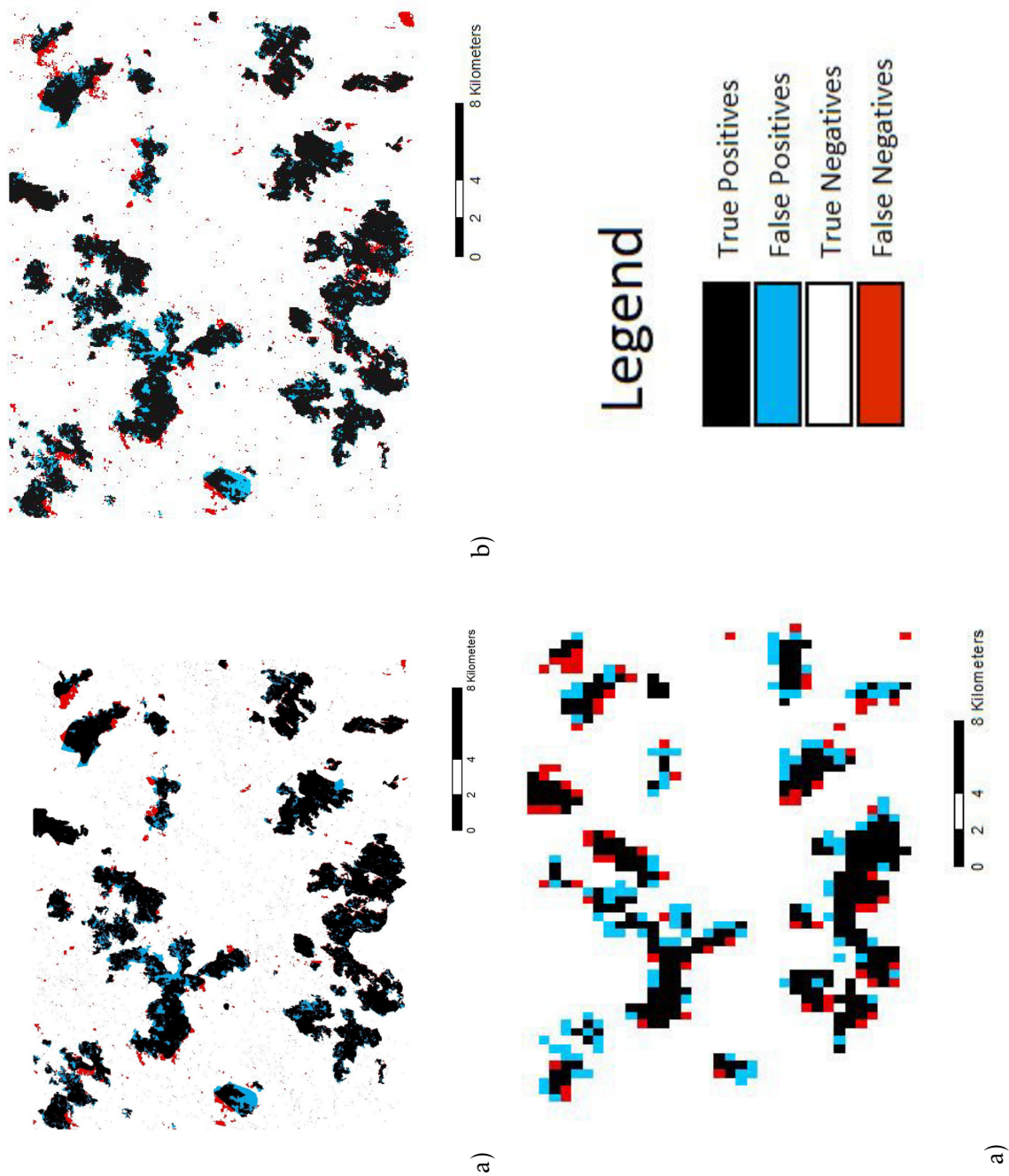


Figure 4.16: Confusion maps generated from a) Sentinel 2 classification b) Landsat 8 classification c) MODIS classification.

In order to further study how the classification results manifest themselves in the study area, a weighted confusion map was implemented, in order to observe the overall opinion of an ensemble composed by different classifiers, voting on the which category should be attributed to a specific pixel. For this intent, the Sentinel 2 dataset was chosen due its greater resolution, as well as the top 5 scoring classifier and training set size combinations for this dataset. The chosen classifiers were:

Classifier	Training set size (%)
XGB	5
MLP	20
liquidSVM	5
GBC	10
KNN	10

This voting approach was chosen because of its capability of generating association ratios for the classification of that pixel belonging to any of the categories of a confusion matrix (TP, TN, FP or FN). The pseudo-code of voting process is in Listing 4.7.6.

Listing 4.1: Voting algorithm

```
1  foreach pixel in raster:
2
3      foreach classifier in ensemble:
4
5          vote = ground truth pixel value - classification pixel value
6
7          if(vote == 1):
8              False Negatives++
9
10         else if(vote == -1):
11             False Positives++
12
13     ratio = (False Negatives - False Positives)/5
```

The ratio generated from the normalization of the difference between each voting class is then discretized into 8 categories that describe the amount of agreement with the ground truth. These categories correspond to numerous classification statements. These can be that the ground truth on that pixel may be classified as burned when it in fact may not be (FP), the opposite (FN) or if the classification is in complete agreement with the category of the ground truth (TP and TN). Table 4.9 demonstrates the relation between the votes of each classifier and their subsequent ratios, and the agreement level between them on which category to choose.

Table 4.9: Discretization of the voting outcome

Votes			Agreement level of relation	Category of relation
FN (1)	FP (-1)	Generated Ratio		
5	0	1	Unanimous	FN
4	1	3/5	Major	FN
3	2	1/5	Borderline	FN
2	3	-1/5	Borderline	FP
1	4	-3/5	Major	FP
0	5	-1	Unanimous	FP
0	0	0	Neutral	TN
1	1	0	Neutral	TP

After the voting process was completed, each ratio was stored in the position of the pixel it related to, and converted into a raster. This method enables the analysis of the agreement between classifiers through a geographically referenced map (see Figure 4.17). At a distance it appears to be identical to confusion maps already presented, but when analysed more closely, the nuances of the different ratios become evident near the borders of the burned areas (see Figure 4.18).

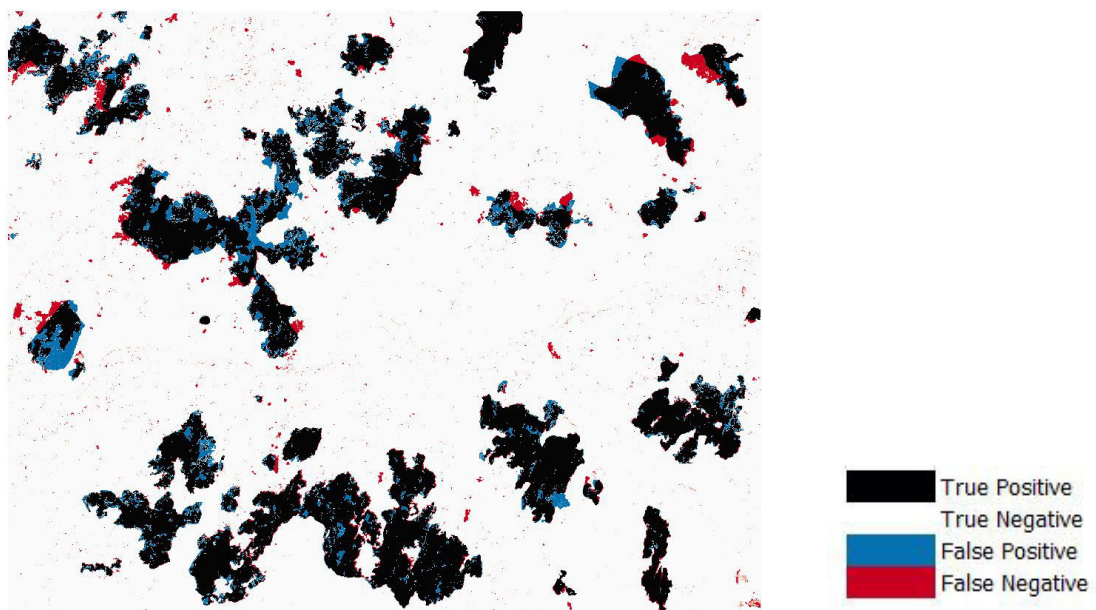


Figure 4.17: a) Weighted Confusion Map

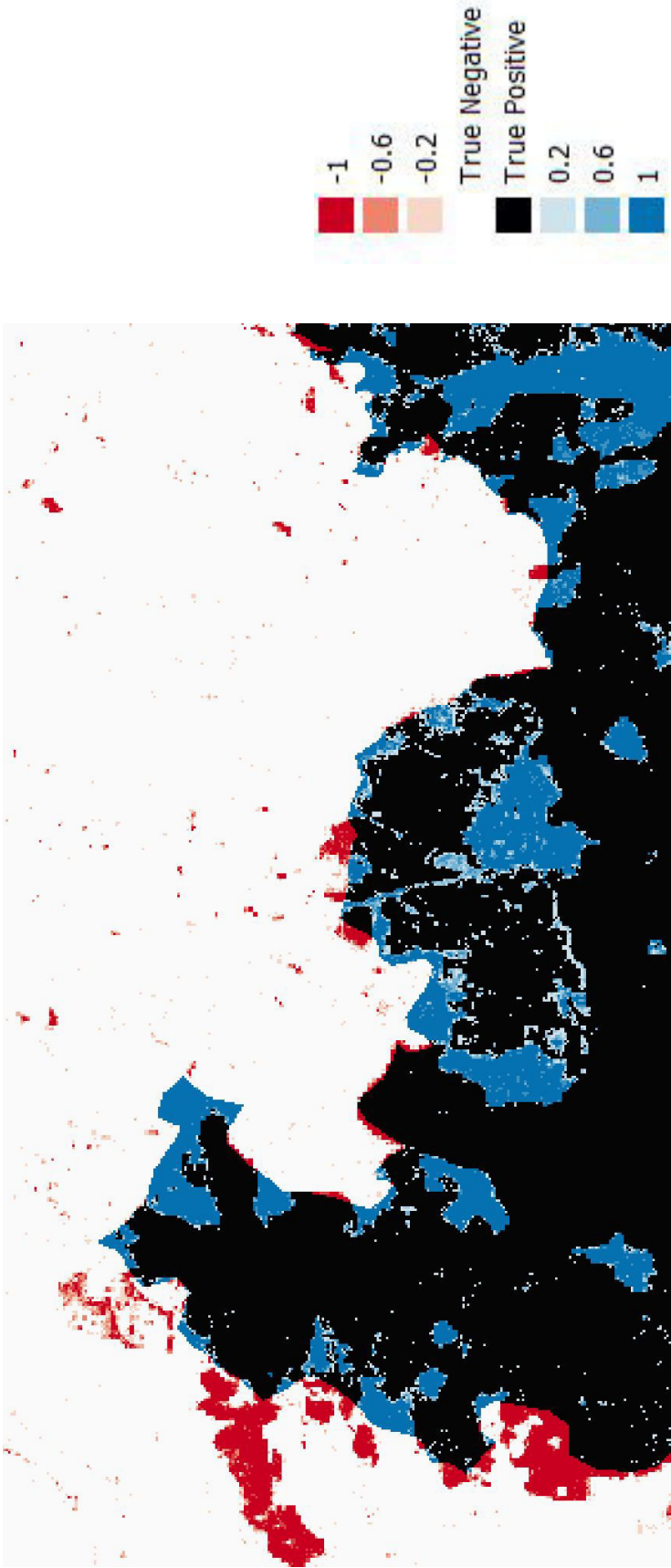


Figure 4.18: Weighted Confusion Map close-up.

4.8 Complementary studies

The aim of these studies is to build upon lessons learned from previously in this study and explore different courses to take from this point.

4.8.1 Fire season burned area classification

The purpose of this experiment is to define a burned area identification for all the days in the fire season (May 1st to October 31st). For this intent we defined our "pre-fire scenario" to the day before the official start of the season (April 30th). The model chosen was a XGBoost classifier trained with 30% of the MODIS dataset and the following parameters: `max_depth = 4`, `min_samples_split = 2` and `n_estimators = 100`.

This choice for combination of classifier, parameters and satellite was based on section 4.7.6. Seeing that this classifier achieved the best overall scores using MODIS data. The choice of MODIS is mainly based on its daily availability of products. The metrics obtained from this process were the following:

Table 4.10: Fire season classification metrics

Operation	Time (s)
Load training data	0.06
Load fire season data	340.33
Total train time	376.05
Total classification time	4119.27
Write classifications	42.9
Total time	4539

Figure 4.19 represents an example of classification of our study area after all the wildfire activity has ended. There appears to be some noise or missclassifications of rock formations, but this can be derived from rock formations being classified as burned, yet the greater burned areas seem to be majorly classified.

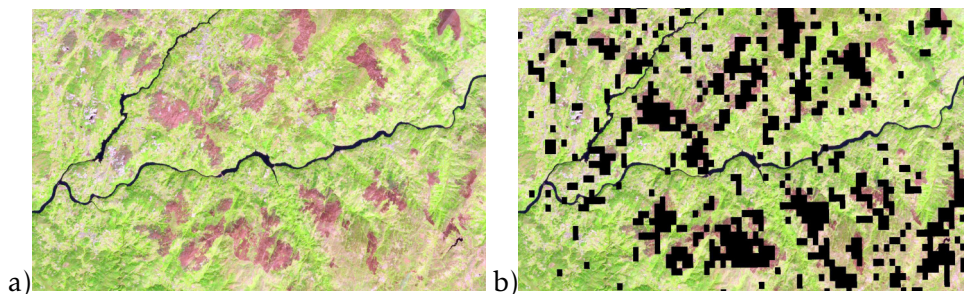


Figure 4.19: a) shows the study area on the 27th of September according to Sentinel 2 and b) shows classification result for 31st of October.

4.8.2 Identification of areas with high risk of fire

This experiment has the aim of inferring areas with high risk of fire occurrence. Since wildfire tends to behave differently according to the meteorological conditions, terrain cover type, characteristics such as elevation and slope as well as the state of the vegetation. A dataset incorporated examples of these features (see Table 4.11) was compiled. All of the referred features were converted from their native resolution to that of MODIS. In order to give the algorithm the most flexibility in order to adjust to this problem, XGBoosts parameters were left at their default values.

Table 4.11: List of features used for risk identification

Multi Spectral Indices	Fire Weather Indices	Land Features
NDVI	FFMC	DEM
SAVI	DMC	Slope
MSI	DC	Distance to nearest road
MIRBI	ISI	Vegetation Covertypes
NBR	BUI	
NBR2	FWI	
NBR3	DSR	
NBR4		
NBR4		

The Fire Weather Indices and Land Features required some processing before use. The first was provided by IPMA and came in text files only with the indice values for the corresponding cells. Yet these values were not geographically referenced, and thus had to be processed into a Comma Separated Values (CSV) applying the algorithm in Listing 4.8.2. After all files were processed, these were converted from CSV to vector files in QGIS that were subsequently rasterized generating a 580 pixels x 540 pixels raster.

Listing 4.2: Fire Weather Indice geographical referencing algorithm

```

1 for line in range(0,fwi.shape[0]):
2     latitude = 44.79 - 0.02 * (line)
3     for col in range(0,fwi.shape[1]):
4         longitude = -12.79 + 0.02 * (col)
5         data = str(latitude)+' ',''+str(longitude)+' ',''+str(fwi[line][col])

```

For the Land Features, the DEM and Vegetation Cover Type were download from the European Environment Agency (EEA) dataset repository (<https://www.eea.europa.eu/data-and-maps>) while the dataset containing all the roads was download from OpenStreetMap (<https://www.openstreetmap.org/>). The Slope feature was created using the Slope tool in QGIS and the Distance to nearest road feature was generated by using

the Proximity tool, also from QGIS.

In order to establish a trend of feature changes over the previous days, we decided to include the Multi Spectral Indices and Fire Weather Indices of the day in question and the two previous days. The Land Features are only used once since they do not tend to change under normal circumstances. To make sure the training labels are not contaminated with already burned areas, these were removed in order to make sure task carried is risk identification and not burned area mapping. In our case, there was a burned area in the center of our study area, hence being ignored. After this the shape was clipped, rasterized and set its resolution to the one of MODIS using a maximum criteria for the sampling function. This function ensures that if the area has a mix of burned and unburned outcomes they are included, thus ensuring that not only "pure" features are taken into account when relating to a burned outcome but also the ones nearby. Figure 4.20 illustrates these final steps.

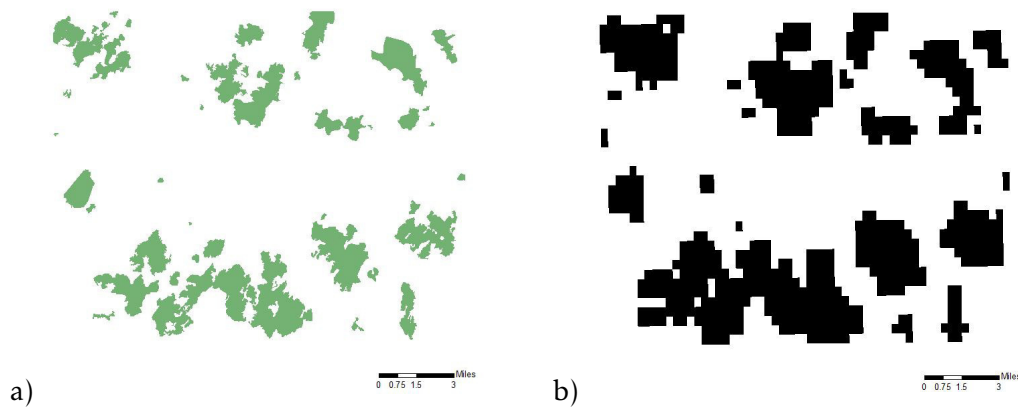


Figure 4.20: a) shows ICNFs burned areas and b) shows the ground truth used.

4.8.2.1 Results

The training data is dated August 1st which includes features from that day as well as the two previous days. The results were surprising since the classifications (August 10th and 11th), resulted in the identification of risk areas near burned areas that the summary provided by the ICNF stated had fire activity in during the month of August. Figure 4.21 illustrates the results obtained.

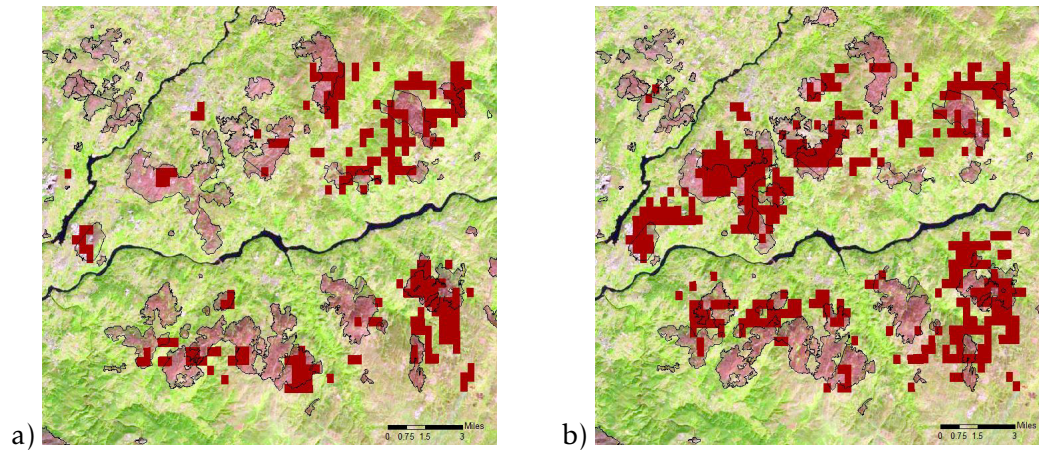


Figure 4.21: a) shows the risk identification for August 10th b) shows the risk identification for August 11th. These classifications are overlaid on the ICNFs summary

After seeing the results and analysing their distribution we assume that there may be some underlying patterns in the features of those areas which relate to an outcome of burned areas seen in the training phase. These results are born as demonstration of the generalization capability of the trained model.

4.8.3 Conclusion

The use of Sentinel 2 and Landsat 8 imagery appears to have some intrinsic value relating to the refinement of existing burned area maps or even its automation due to the relative ease of features generated and total execution time some algorithms take. Sentinel 2 when trying classify the entire extent covered by product (cell), it proved to a task to heavy memory-wise to handle by the machine used in this experiment. On the other hand, Landsat 8 albeit its slower revisit rate possesses a resolution decrease by a factor large enough to make the size of the dataset generated more manageable but small enough to not lose to much quality. Yet Landsat 8's use comes with the caveat of having some increase in the noise with its imagery probably associated with the resolution decrease. The complementary experiments displayed above demonstrate us that the use MODIS imagery for daily burned area identification poses as possibility of generating on-the-fly preliminary burned area risk classifications that may prove useful in support to decision making when pondering strategies for wildfire management and prevention.

CONCLUSION & FUTURE WORK

This chapter aims on drawing conclusions as well as giving recommendations on how further studies based on this work should proceed.

5.1 Conclusion

In summary, seven classifiers were the subjects of this study, which consisted on the head-to-head comparison of the standard versions of these classifiers with their accelerated counterparts. In order to evaluate their performance, they were used in a classification task for burned area identification in the Castelo de Paiva region. As source for the dataset, Sentinel 2, Landsat 8 and MODIS imagery was used in order to verify their performance with different dataset sizes and resolutions. The best results obtained with each classifier was:

1. XGB Accuracy = 0.96 and Kappa = 0.86 in 4.59 seconds
2. MLP Accuracy = 0.96 and Kappa = 0.86 in 80.68 seconds
3. GBC Accuracy = 0.96 and Kappa = 0.86 in 102.13 seconds
4. liqSVM Accuracy = 0.96 and Kappa = 0.86 in 626.14 seconds
5. KNN Accuracy = 0.95 and Kappa = 0.85 in 126.98 seconds
6. SVC Accuracy = 0.95 and Kappa = 0.85 in 6183.29 seconds
7. DNN Accuracy = 0.94 and Kappa = 0.82 in 107.45 seconds

When looking at the overall assessment metric values, results seem promising derived from the unbalanced nature of the dataset corresponding to our area of study, since in

55900 ha there are 44100 ha (79%) that are not burned and 11800 ha that are burned (21%). This means that samples representing burned areas are less represented when using a stratified split and possibly nonexistent otherwise. Having stated this, the overall results were good, having the best results achieved accuracies of 94% to 96% mark. The good nature of these results may derive, but are not limited, from:

1. Sufficient sample amount;
2. Sentinel 2's image quality
3. The variety of features, both of Sentinel 2 bands and indices, when performing feature selection.

It is our opinion that if these key-points were addressed the results could see an increase:

1. Better balancing of the class distribution;
2. A more refined ground truth;

5.1.1 Classifiers

If only looking at Accuracy and Kappa, we can state that the XGB, liqSVM, MLP and GB classifiers achieved the top scores (96% Accuracy) for the task at hand. albeit the remainder of the classifiers being close behind with the SVC and KNN achieving 95%, with the latter being noteworthy for such a simple classifier being on par with more complex classifiers.

If we now look at the execution times, we can see that XGBoost dominates in matters of overall execution times and accuracy assessment metrics. It takes nearly 7 seconds while the quickest classifier after it takes 77 seconds. In comparison to its counterpart, the GB classifier, XGBoost outperforms it with 32-fold execution time speedup while maintain the same Accuracy and Kappa scores. It also worthy to mention the significant execution time decrease of the liquidSVM classifier in comparison to its counterpart (SVC) which is 10 times slower.

In summary, the use of accelerated classifiers can be worthwhile due to their reduced execution times that offsets their small to none accuracy reduction. In other cases, some are still in early development, like the DNN classifier and the entire Tensorflow library, leaving us to wait for what performance increases may come with future releases.

5.1.2 Satellites

By analysing the results, we find that the best of them were obtained by using the Sentinel 2 dataset. This is derived from its great resolution (10m x 10m) which converts to smaller extents of land and an immense amount of samples that can be used to overview the terrain with great detail. On the other hand we have [MODIS](#) that has a smaller resolution

(250m x 250m) that abstracts the terrain features by a huge amount but has the benefit of generating tiny datasets, in comparison to both Sentinel 2 and Landsat 8's dataset in relation to the same terrain extent, that classifiers who take advantage from small amounts of data like [SVMs](#) and are disadvantageous in other scenarios. [MODIS](#) also provides wide product range with daily availability, in opposition to the Sentinels 5-day and Landsats 16-day revisit frequency. As a compromise between resolution and dataset size, Landsat 8 provides good resolution products but without weighing as much as Sentinel 2 data that in some experiments resulted in problems with insufficient memory allocation. Yet, it brings the caveat of only generating products for the same grid cell every 16 days.

5.2 Future Work

In order to ease the decision of how to proceed with this study we propose the following aspects:

More classes Burned area identification is usually used over large extents, thus incorporating a diverse range of land cover types. Classification may improve if knowledge of different cover types is incorporated leading to a possible reduction in wrong classifications that share the same spectral signature as the burned areas.

Smaller time intervals Using closer gaps to the fire activity can help define how the burned areas change closely before and after the event.

Historical context The use past fire events and how they relate with feature changes can help train a more generalized model to be used across wider areas and different times of the year. This opinion is derived from the state of deterioration of the vegetation is different for different regions at different times of the year. Thus, having some weight when performing a country-wide classification procedure.

Object-based approach Since burned areas have similar values between neighbouring pixels in addition to pixels with identical labels appearing across wide continuous areas, the classification process may benefit from an object based approach instead of a pixel-based one.

Optimized Python distribution Intel has been marketing its "accelerated" Python distribution that claim to boost performance when running on [CPU's](#). This can pose as possibility to study performance gains of using this distribution on high-end [CPUs](#) like Intel Xeons or Intel Phi standalone or also combined with [GPU](#)-accelerated algorithms reducing any existing bottleneck in performance.

BIBLIOGRAPHY

- [1] E. S. Agency. *Copernicus. Observing the earth*. 2017. URL: <http://www.esa.int>.
- [2] E. Alpaydin. *Introduction to machine learning*. 2nd ed. isbn: 978-0-262-01243-0. Massachusetts Institute of Technology: The MIT Press, 2010.
- [3] A. Arpacı, B. Malowerschnig, O. Sass, and H. Vacik. “Using multi variate data mining techniques for estimating fire susceptibility of Tyrolean forests.” In: *Applied Geography* 53 (2014), pp. 258 –270. ISSN: 0143-6228. DOI: <http://dx.doi.org/10.1016/j.apgeog.2014.05.015>. URL: <http://www.sciencedirect.com/science/article/pii/S0143622814001106>.
- [4] author. *RELATÓRIO PROVISÓRIO DE INCÊNDIOS FLORESTAIS (2017). 01 DE JANEIRO A 31 DE OUTUBRO*. ICNF.
- [5] BAER. BARC. US Forest Service. 2017. URL: <https://www.fs.fed.us/eng/rsac/baer/barc.html>.
- [6] K. Barrett, E. Kasischke, A. McGuire, M. Turetsky, and E. Kane. “Modeling fire severity in black spruce stands in the Alaskan boreal forest using spectral and non-spectral geospatial data.” In: *Remote Sensing of Environment* (114), 1494–1503. DOI: <https://doi.org/10.1016/j.rse.2010.02.001>. URL: <http://www.sciencedirect.com/science/article/pii/S003442571000060X>.
- [7] *Base de Dados Portugal Contemporâneo*. URL: <http://www.pordata.pt/>.
- [8] I. Bekeryte. *Satellite Systems - History, Definition, Functioning Principles and Application Spheres*. 2007.
- [9] N. R. Canada. *Satellite Imagery and Products. Educational Resources*. Government of Canada. URL: <http://www.nrcan.gc.ca/node/9309>.
- [10] COPERNICUS. *European Forest Fire Information System*. 2018. URL: <http://copernicus.eu/main/european-forest-fire-information-system>.
- [11] COPERNICUS and EC. *Emergency Management Service*. COPERNICUS and EC. 2018. URL: <http://effis.jrc.ec.europa.eu/>.

- [12] M. Drusch, U. D. Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, A. Meygret, F. Spoto, O. Sy, F. Marchese, and P. Bargellini. “Sentinel-2: ESA’s Optical High-Resolution Mission for GMES Operational Services.” In: *Remote Sensing of Environment* 120 (2012). The Sentinel Missions - New Opportunities for Science, pp. 25 –36. ISSN: 0034-4257. DOI: <http://dx.doi.org/10.1016/j.rse.2011.11.026>. URL: <http://www.sciencedirect.com/science/article/pii/S0034425712000636>.
- [13] GDAL. *Sentinel-2 Products*. 2017. URL: http://www.gdal.org/frmt_sentinel2.html.
- [14] J. Ghorpade, J. Parande, M. Kulkarni, and A. Bawaskar. “GPGPU PROCESSING IN CUDA ARCHITECTURE.” In: *Advanced Computing: An International Journal* (2012). <https://ai2-s2-pdfs.s3.amazonaws.com/9e23/d532e25b14b3760dc6d410b3d0bfeabeb91c.pdf>.
- [15] ICNF. *Incêndios Florestais*. 2018. URL: <http://www2.icnf.pt/portal/florestas/dfci/inc/mapas>.
- [16] ICNF, ANPC, CNOS, GNR, and DGACPPF. *Relatório Anual de Áreas Ardidas e Incêndios Florestais em Portugal Continental*. Yearly Report. ICNF. 46 pp. URL: <http://www.icnf.pt/portal/florestas/dfci/Resource/doc/rel/2015/ICNF-Relatorio-Anual-Inc-2015-v2017mar23.pdf>.
- [17] R. Inam. *An Introduction to GPGPU Programming - CUDA Architecture*. Tech. rep. 2010. URL: <http://www.es.mdh.se/publications/1994->.
- [18] IPMA. *Enciclopédia*. 2017. URL: <https://www.ipma.pt/pt/enciclopedia>.
- [19] IPMA and AEMet. *Iberian Climate Atlas. Air Temperature and Precipitation (1971-2000)*. ES, PT, and EN. Ed. by A. E. de Meteorología. Closas-Orcoyen S. L., 2011, 29 –30 and 63 –65. 80 pp. ISBN: 978-84-7837-079-5. URL: http://www.ipma.pt/resources.www/docs/publicacoes.site/atlas_clima_iberico.pdf.
- [20] ISA. *CLIMA, METEOROLOGIA E INDEXAÇÃO METEOROLÓGICA*. 2018.
- [21] A. Karouni, B. Daya, and P. Chauvet. “Applying decision tree algorithm and neural networks to predict forest fires in Lebanon.” In: *Journal of Theoretical and Applied Information Technology* 63.2 (). ISSN: 1992-8645. URL: https://www.researchgate.net/publication/287863331_Applying_decision_tree_algorithm_and_neural_networks_to_predict_forest_fires_in_Lebanon.
- [22] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih. “PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation.” In: *Parallel Computing* 38.3 (2012), pp. 157–174. ISSN: 0167-8191. DOI: [10.1016/j.parco.2011.09.001](https://doi.org/10.1016/j.parco.2011.09.001).

- [23] L. Krippahl. *Aprendizagem Automática (Machine Learning). Lecture Notes*. Departamento de Informatica, Universidade Nova de Lisboa. 2016. URL: <http://aa.ssdi.di.fct.unl.pt/>.
- [24] D. Lutes, R. Keane, J. Caratti, N. B. C. Key, S. Sutherland, and L. Gangi. *Fire Effects Monitoring and Inventory System. Integration of Standardized Field Data Collection Techniques and Sampling Design With Remote Sensing to Assess Fire Effects*. Tech Report. US Department of Agriculture, US Forest Service, and US Department of the Interior.
- [25] J. Maurer. *Overview of NASA's Terra satellite*. Honolulu, USA. University of Hawaii at Manoa. URL: <http://www2.hawaii.edu/~jmaurer/terra/>.
- [26] A Mazher. "Comparative analysis of mapping burned areas from landsat TM images." In: *Journal of Physics*. 439th ser. (2013). DOI: 10.1088/1742-6596/439/1/012038. URL: <http://iopscience.iop.org/article/10.1088/1742-6596/439/1/012038/pdf>.
- [27] J. D. Miller, E. E. Knapp, C. H. Key, C. N. Skinner, C. J. I. R., M. Creasy, and J. W. Sherlock. "Calibration and validation of the relative differenced Normalized Burn Ratio (RdNBR) to three measures of fire severity in the Sierra Nevada and Klamath Mountains, California, USA." In: *Remote Sensing of Environment* (113 2009), 645–656. DOI: doi:10.1016/j.rse.2008.11.009.
- [28] G. Mountrakis, J. Im, and C. Ogole. "Support vector machines in remote sensing: A review." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.3 (2011), pp. 247–259. ISSN: 0924-2716. DOI: <http://dx.doi.org/10.1016/j.isprsjprs.2010.11.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0924271610001140>.
- [29] H. Naganathan, S. P. Seshasayee, J. Kim, W. K. Chong, and J.-S. Chou. "Wildfire Predictions: Determining Reliable Models using Fused Dataset." Version version. In: *Global Journal of Computer Science and Technology: C Software Data Engineering*. series 16 (4 2016), pages. ISSN: 0975-4172. URL: https://www.researchgate.net/publication/310481821_Wildfire_Predictions_Determining_Reliable_Models_using_Fused_Dataset. addendum. pubstate.
- [30] NASA. *What are passive and active sensors?* Ed. by T. Mai. URL: https://www.nasa.gov/directorates/heo/scan/communications/outreach/funfacts/txt_passive_active.html.
- [31] NASA. MODIS. *Moderate Resolution Spectoradiometer*. 2017. URL: <https://modis.gsfc.nasa.gov/>.
- [32] NASA. *Measuring Vegetation*. 2018. URL: https://earthobservatory.nasa.gov/Features/MeasuringVegetation/measuring_vegetation_2.php.

- [33] NRCan. *Canadian Wildland Fire Information System*. Government of Canada. URL: <http://cwfis.cfs.nrcan.gc.ca/home>.
- [34] R. C. Nunes, C. V. Damásio, S. Nascimento, and A. Sousa. “Automatic Crop Classification in Alentejo Region using Landsat-8 vs Sentinel Imagery.” Masters Thesis. Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, 2017.
- [35] M. Pantic. *Machine Learning (course 395)*. Imperial College of London. 2017. URL: <https://ibug.doc.ic.ac.uk/media/uploads/documents/courses/ml-lecture4.pdf>.
- [36] S. A. Parks, G. K. Dillon, and C. Miller. “A New Metric for Quantifying Burn Severity: The Relativized Burn Ratio.” In: *Remote Sensing* 6 (2014), pp. 1827–1844. ISSN: ISSN 2072-4292. DOI: [doi:10.3390/rs6031827](https://doi.org/10.3390/rs6031827).
- [37] G. P. Petropoulos, C. Kontoes, and I. Keramitsoglou. “Burnt area delineation from a uni-temporal perspective based on Landsat TM imagery classification using Support Vector Machines.” In: *International Journal of Applied Earth Observation and Geoinformation* 13 (1), pp. 70–80. DOI: <https://doi.org/10.1016/j.jag.2010.06.008>.
- [38] P. D. M. Riedmiller. *Machine Learning. Multi Layer Perceptrons*. Albert-Ludwigs-University Freiburg. 2010. URL: http://ml.informatik.uni-freiburg.de/_media/teaching/ss10/05_mlps.printer.pdf.
- [39] G. Rondeaux, M. Steven, and F. Baret. “Optimization of Soil-Adjusted Vegetation Indices.” In: *REMOTE SENS. ENVIRON* (1996).
- [40] Y. Safi and A. Bouroumi. *A Neural Network Approach for Predicting Forest Fires*. Multimedia Computing and Systems (ICMCS), 2011 International Conference. Casablanca, Morocco: Ben M’sik Faculty of Sciences, Hassan II Mohammedia-Casablanca University, UH2MC. DOI: [10.1109/ICMCS.2011.5945716](https://doi.org/10.1109/ICMCS.2011.5945716). URL: <http://ieeexplore.ieee.org/document/5945716/>.
- [41] Y. Safi and A. Bouroumi. “Prediction of Forest Fires Using Artificial Neural Networks.” In: *Applied Mathematical Sciences* 7.6 (2013), pp. 271–286. DOI: [10.12988/ams.2013.13025](https://doi.org/10.12988/ams.2013.13025). URL: https://www.researchgate.net/publication/290160237_Prediction_of_forest_fires_using_Artificial_neural_networks.
- [42] G. E. Sakr, I. H. Elhajj, G. Mitri, and U. C. Wejinya. “Artificial intelligence for forest fire prediction.” In: *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2010, pp. 1311–1316. DOI: [10.1109/AIM.2010.5695809](https://doi.org/10.1109/AIM.2010.5695809). URL: <http://ieeexplore.ieee.org/document/5695809/>.

-
- [43] G. E. Sakr, I. H. Elhajj, and G. Mitri. "Efficient forest fire occurrence prediction for developing countries using two weather parameters." In: *Engineering Applications of Artificial Intelligence* 24.5 (2011), pp. 888–894. ISSN: 0952-1976. DOI: <http://dx.doi.org/10.1016/j.engappai.2011.02.017>. URL: <http://www.sciencedirect.com/science/article/pii/S0952197611000418>.
 - [44] SATELLITES ARE TRACING EUROPE'S FOREST FIRE SCARS. URL: http://www.esa.int/Our_Activities/Observing_the_Earth/Satellites_are_tracing_Europe_s_forest_fire_scars.
 - [45] L. Schepers, B. Haest, S. Veraverbeke, T. Spanhove, J. V. Borre, and R. Goossens. "Burned Area Detection and Burn Severity Assessment of a Heathland Fire in Belgium Using Airborne Imaging Spectroscopy (APEX)." In: *Remote Sensing* 6 (2014), pp. 1803–1826. DOI: [doi:10.3390/rs6031803](https://doi.org/10.3390/rs6031803).
 - [46] D. Stojanova, A. Kobler, P. Ogrinc, B. Ženko, and S. Džeroski. "Estimating the risk of fire outbreaks in the natural environment." In: *Data Mining and Knowledge Discovery* 24.2 (2012), pp. 411–442. ISSN: 1573-756X. DOI: [10.1007/s10618-011-0213-2](https://doi.org/10.1007/s10618-011-0213-2). URL: <http://dx.doi.org/10.1007/s10618-011-0213-2>.
 - [47] S. Tariq. "An Introduction to GPU Computing and CUDA Architecture." In: *GTC On-demand*. GTC Express. http://on-demand.gputechconf.com/gtc-express/2011/presentations/GTC_Express_Sarah_Tariq_June2011.pdf. NVIDIA Corporation. GPU Tech Conference, 2011.
 - [48] S. Trigg and S. Flasse. "An evaluation of different bi-spectral spaces for discriminating burned shrub-savannah." In: *International Journal of Remote Sensing* (2001), pp. 2641–2647. ISSN: ISSN: 0143-1161 (Print) 1366-5901 (Online). DOI: [DOI: 10.1080/01431160110053185](https://doi.org/10.1080/01431160110053185).
 - [49] USGS. *FIREMON BR Cheat Sheet*. Version 4. URL: https://burnseverity.cr.usgs.gov/pdfs/lav4_br_cheatsheet.pdf.
 - [50] USGS. *What are the band designations for the Landsat satellites?* 2017. URL: <https://landsat.usgs.gov/what-are-band-designations-landsat-satellites>.
 - [51] USGS and EROS. *Landsat—Earth observation satellites. Fact Sheet*. English. USGS Numbered Series 2015–3081. Version 1.1. Reston, VA: U.S. Geological Survey. 4 pp. ISBN: 2327-6932. DOI: <http://dx.doi.org/10.3133/fs20153081>. URL: <https://pubs.er.usgs.gov/publication/fs20153081>.
 - [52] A. J. Viera and J. M. Garrett. "Understanding Interobserver Agreement. The Kappa Statistic." In: *Family Medicine* 37 (5), pp. 360–363. URL: <http://www.stfm.org/FamilyMedicine/Vol37Issue5/Viera360>.
 - [53] P. Welikhe, J. Essamuah–Quansah, S. Fall, and W. McElhenney. "Estimation of Soil Moisture Percentage Using LANDSAT-based Moisture Stress Index." In: *Journal of Remote Sensing GIS Journal* (2017). DOI: [DOI: 10.4172/2469-4134.1000200](https://doi.org/10.4172/2469-4134.1000200).



EXECUTION TIMES IN DETAIL

A.1 Multilayer Perceptron Classifier

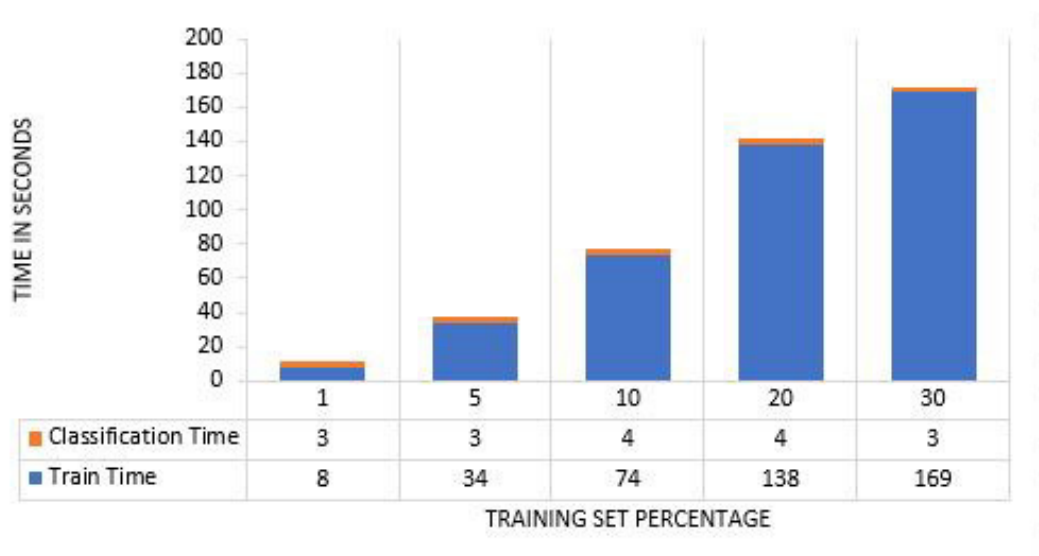


Figure A.1: Execution time detailing training and classification time for the MLP classifier using Sentinel 2 data

APPENDIX A. EXECUTION TIMES IN DETAIL

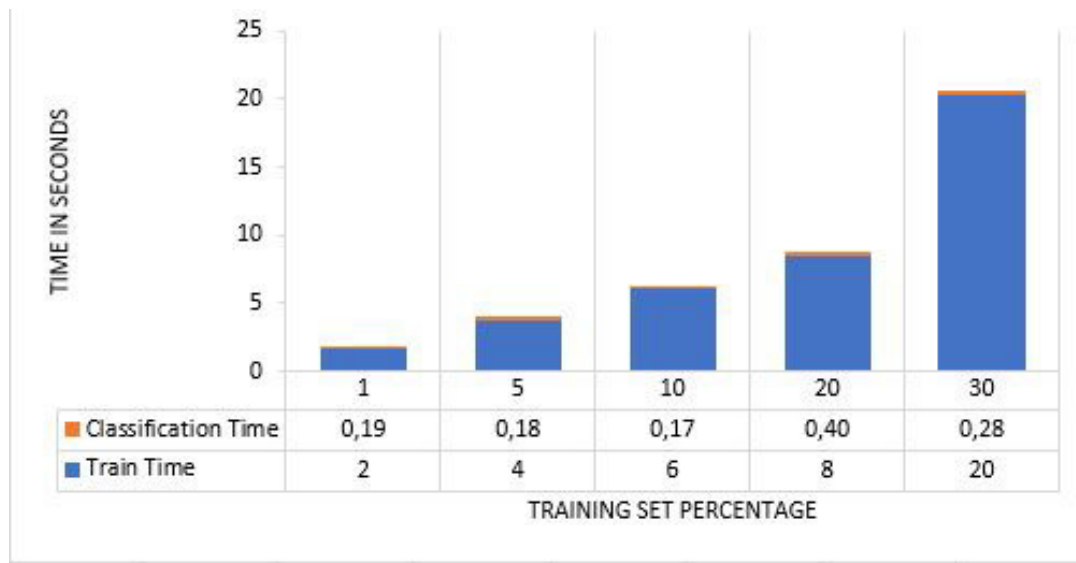


Figure A.2: Execution time detailing training and classification time for the MLP classifier using Landsat 8 data

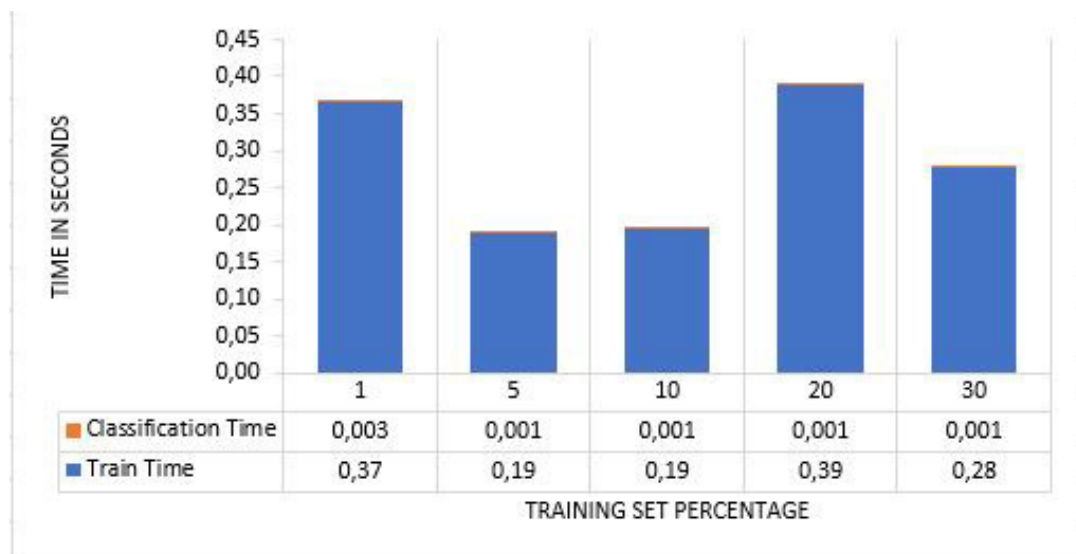


Figure A.3: Execution time detailing training and classification time for the MLP classifier using MODIS data

A.2 Deep Neural Network Classifier

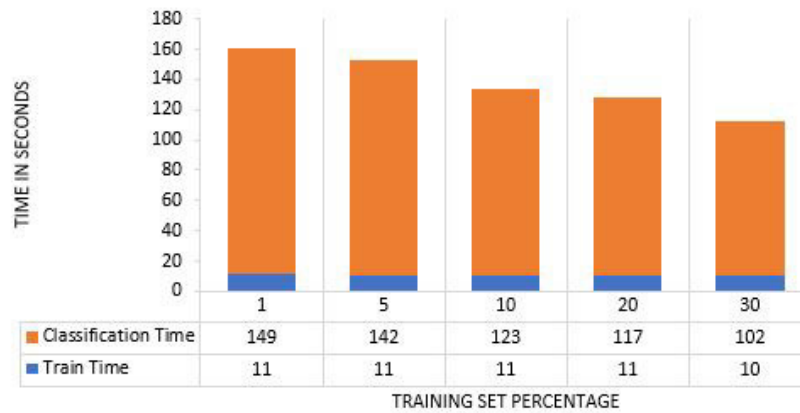


Figure A.4: Execution time detailing training and classification time for the DNN classifier using Sentinel 2 data

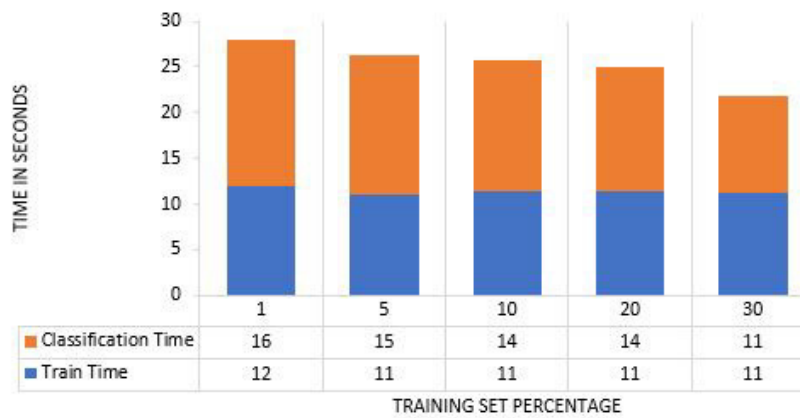


Figure A.5: Execution time detailing training and classification time for the DNN classifier using Landsat 8 data

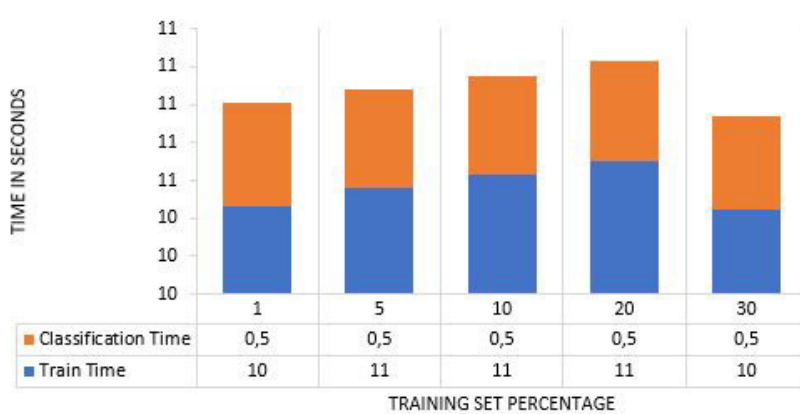


Figure A.6: Execution time detailing training and classification time for the DNN classifier using MODIS data

A.3 Gradient Boosting Classifier

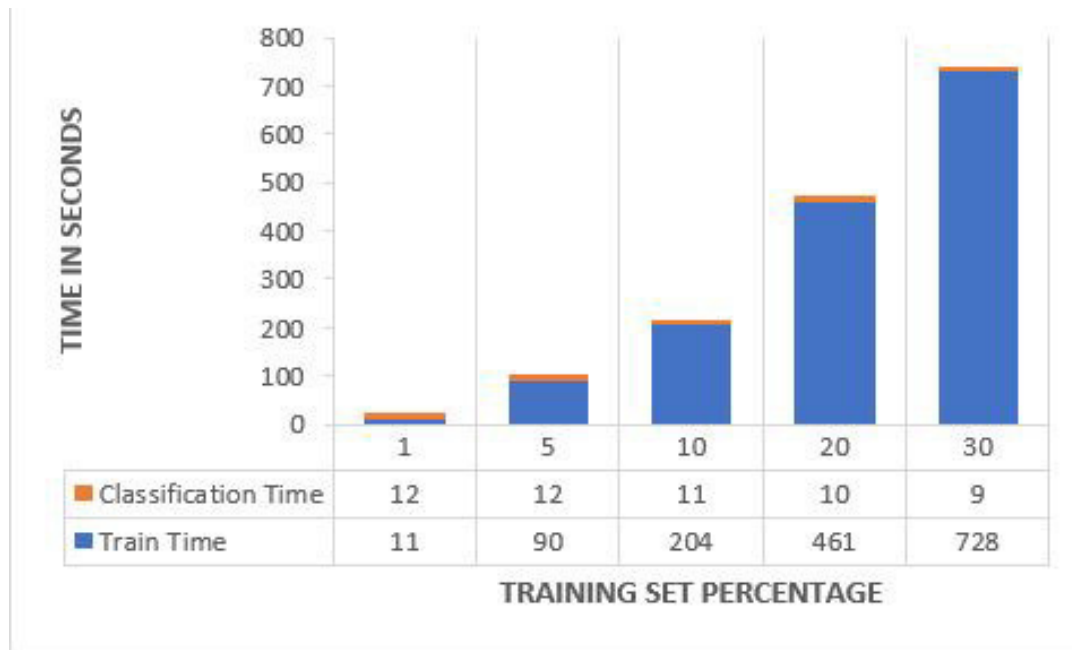


Figure A.7: Execution time detailing training and classification time for the GBC classifier using Sentinel 2 data

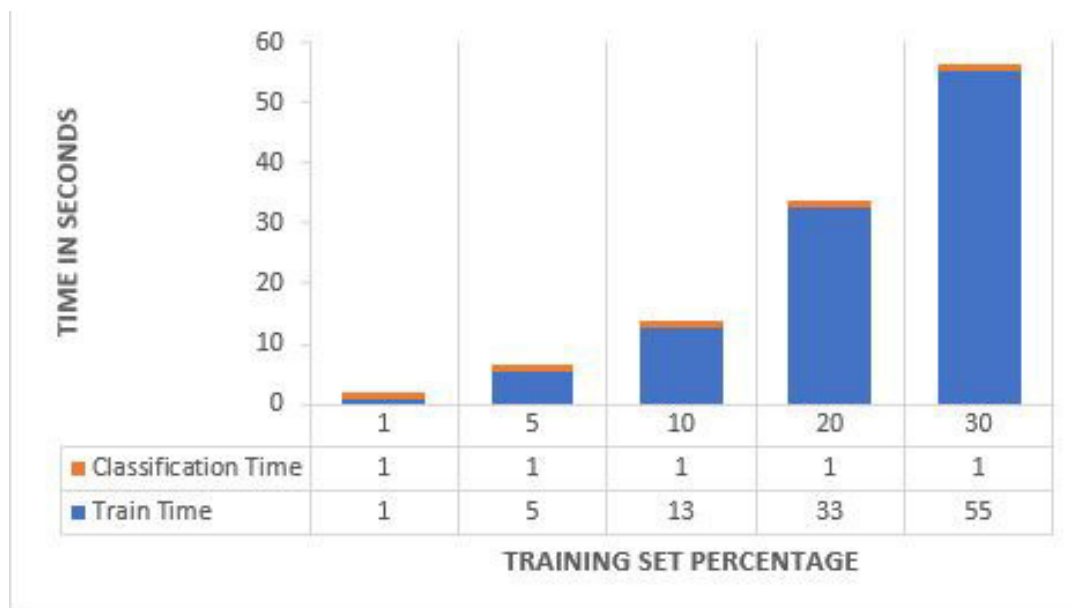


Figure A.8: Execution time detailing training and classification time for the GBC classifier using Landsat 8 data

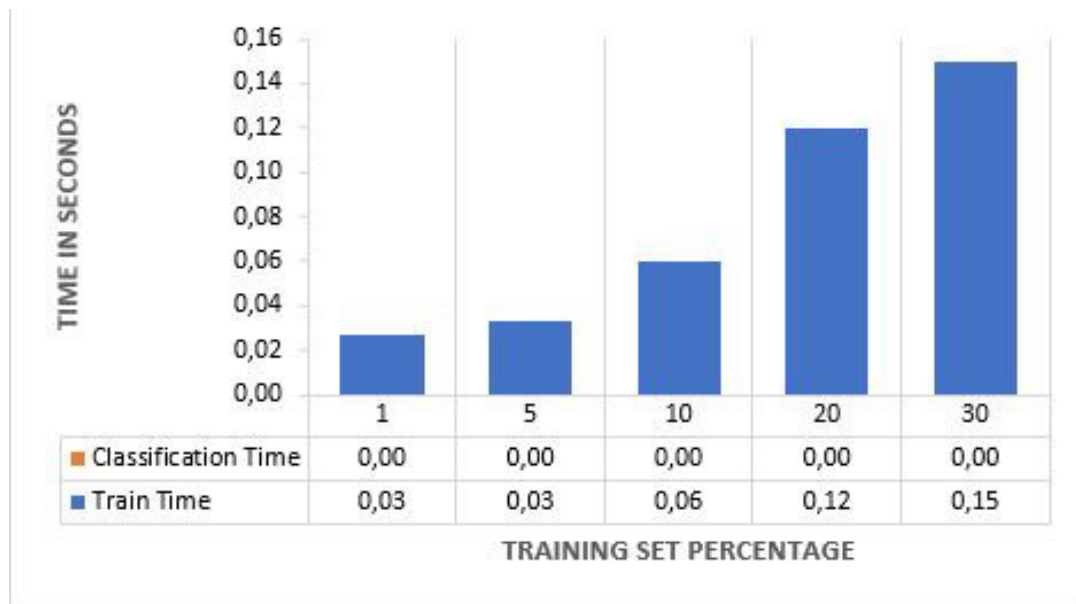


Figure A.9: Execution time detailing training and classification time for the GBC classifier using MODIS data

A.4 XGBoost Classifier

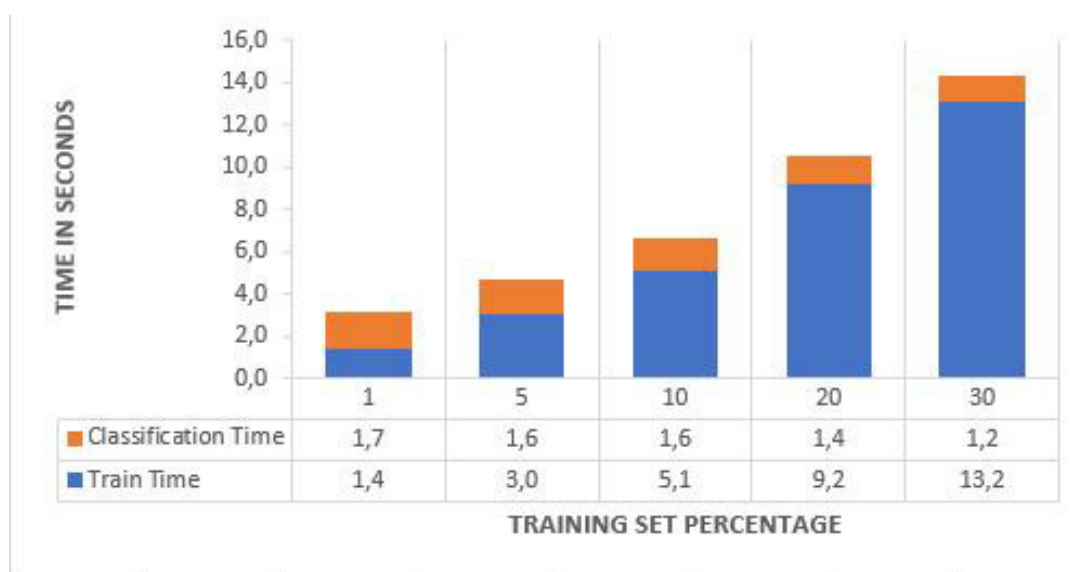


Figure A.10: Execution time detailing training and classification time for the XGB classifier using Sentinel 2 data

APPENDIX A. EXECUTION TIMES IN DETAIL

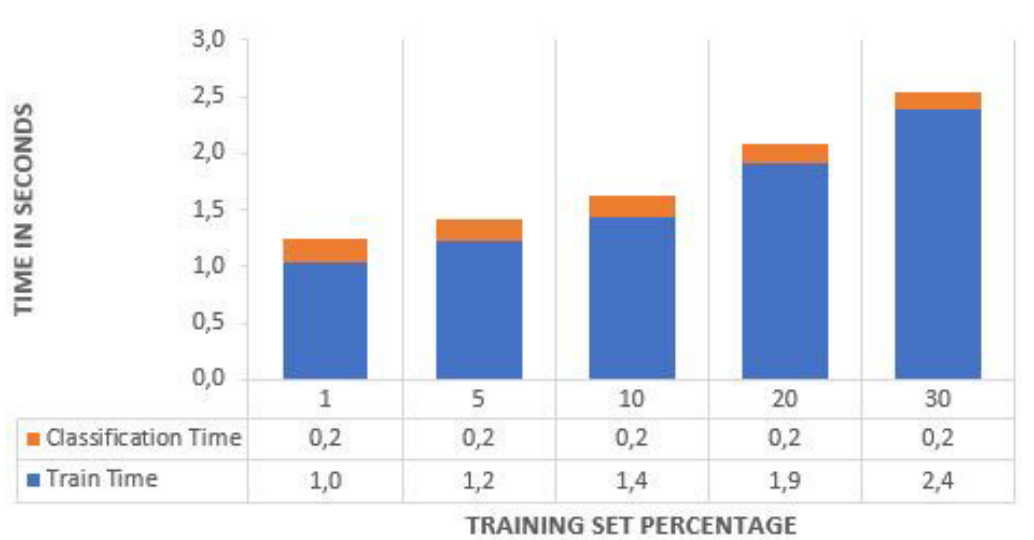


Figure A.11: Execution time detailing training and classification time for the XGB classifier using Landsat 8 data

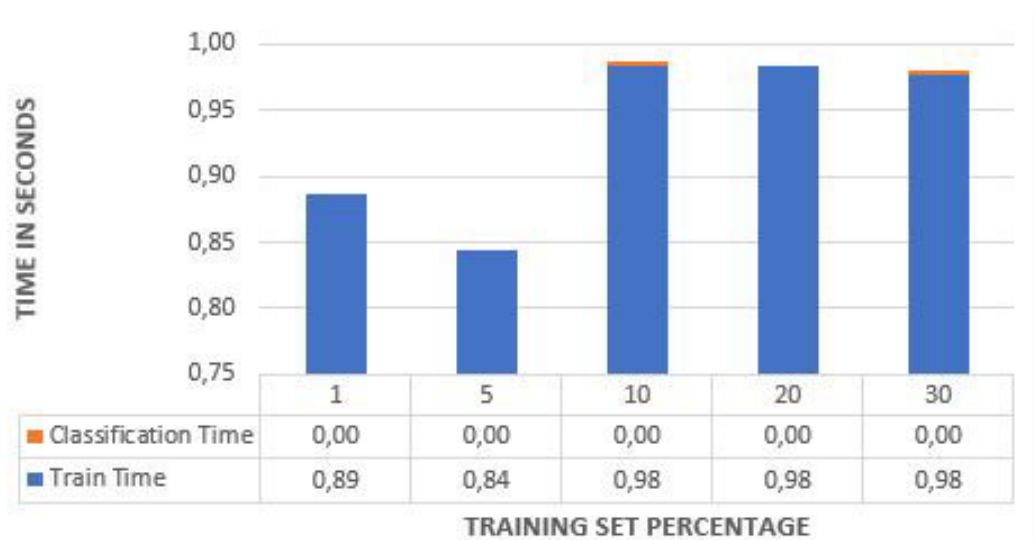


Figure A.12: Execution time detailing training and classification time for the XGB classifier using MODIS data

A.5 Support Vector Classifier

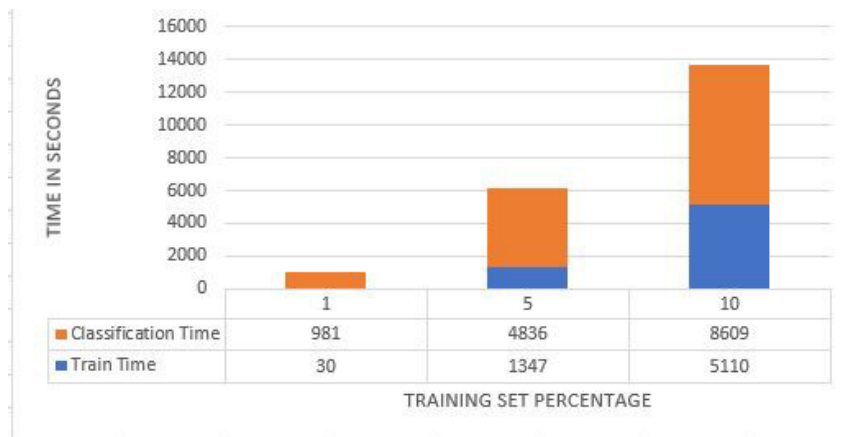


Figure A.13: Execution time detailing training and classification time for the SVC classifier using Sentinel 2 data



Figure A.14: Execution time detailing training and classification time for the SVC classifier using Landsat 8 data

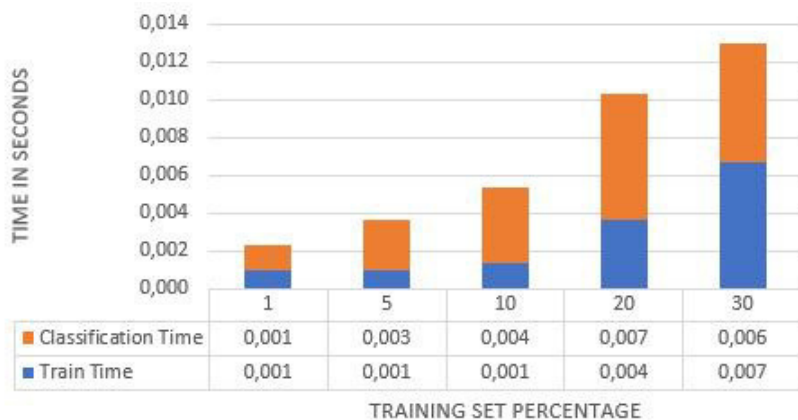


Figure A.15: Execution time detailing training and classification time for the SVC classifier using MODIS data

A.6 liquidSVM Classifier

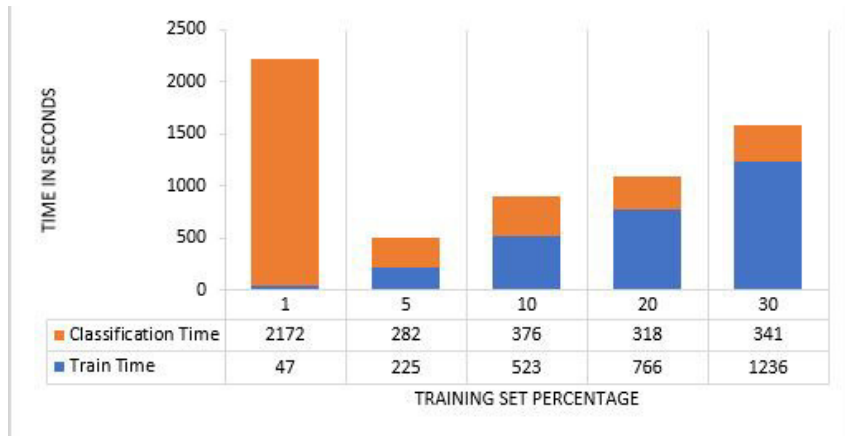


Figure A.16: Execution time detailing training and classification time for the liquidSVM classifier using Sentinel 2 data

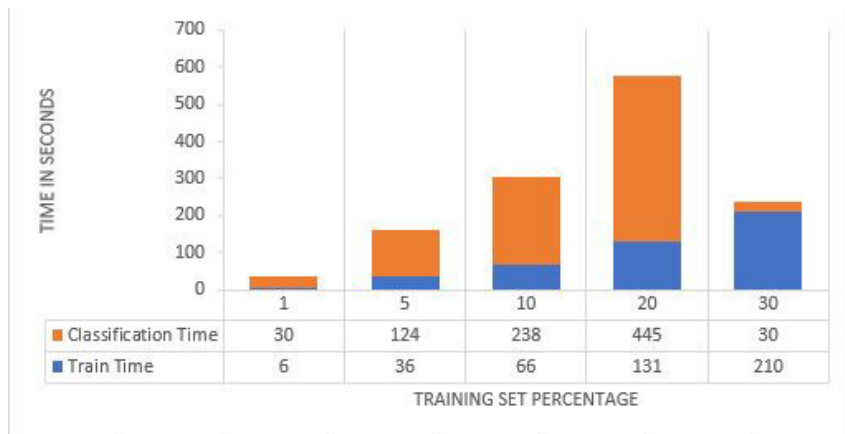


Figure A.17: Execution time detailing training and classification time for the liquidSVM classifier using Landsat 8 data

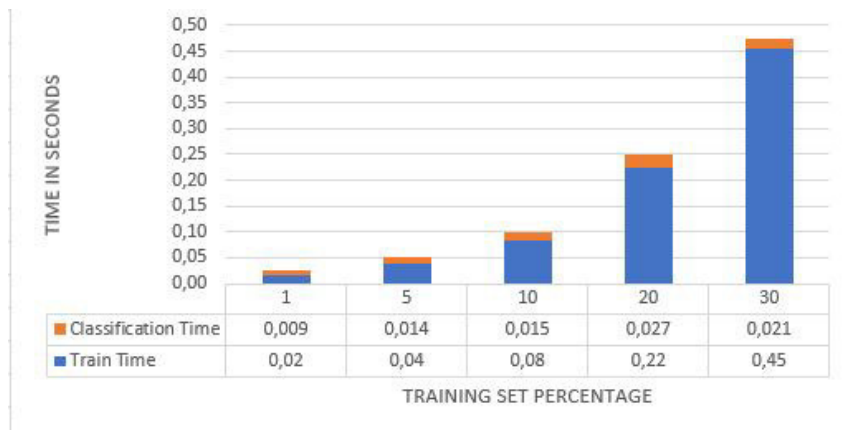


Figure A.18: Execution time detailing training and classification time for the liquidSVM classifier using MODIS data

A.7 K Nearest Neighbours Classifier

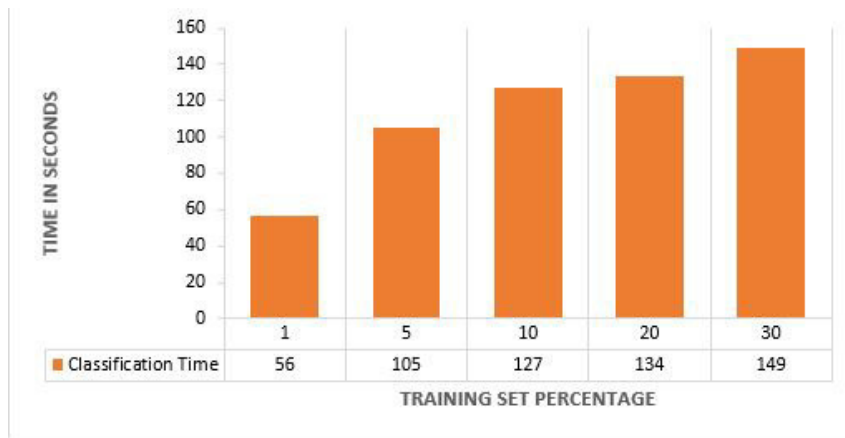


Figure A.19: Execution time detailing training and classification time for the KNN classifier using Sentinel 2 data

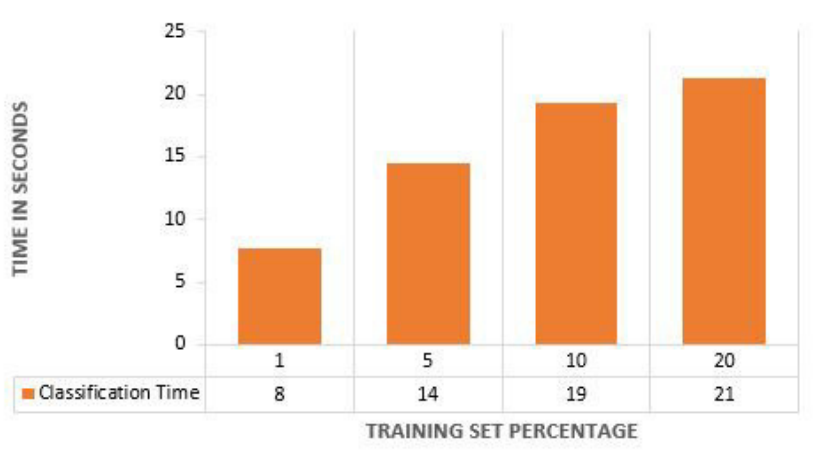


Figure A.20: Execution time detailing training and classification time for the KNN classifier using Landsat 8 data

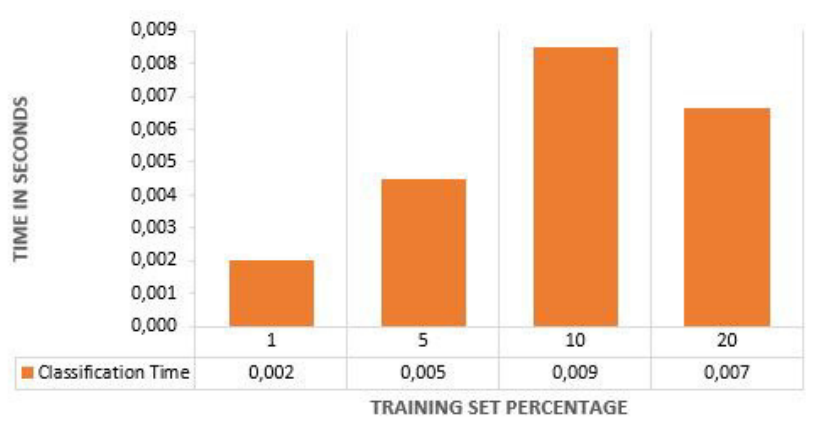


Figure A.21: Execution time detailing training and classification time for the KNN classifier using MODIS data

CLASSIFIER COUNTERPART COMPARISON

B.1 Neural Networks

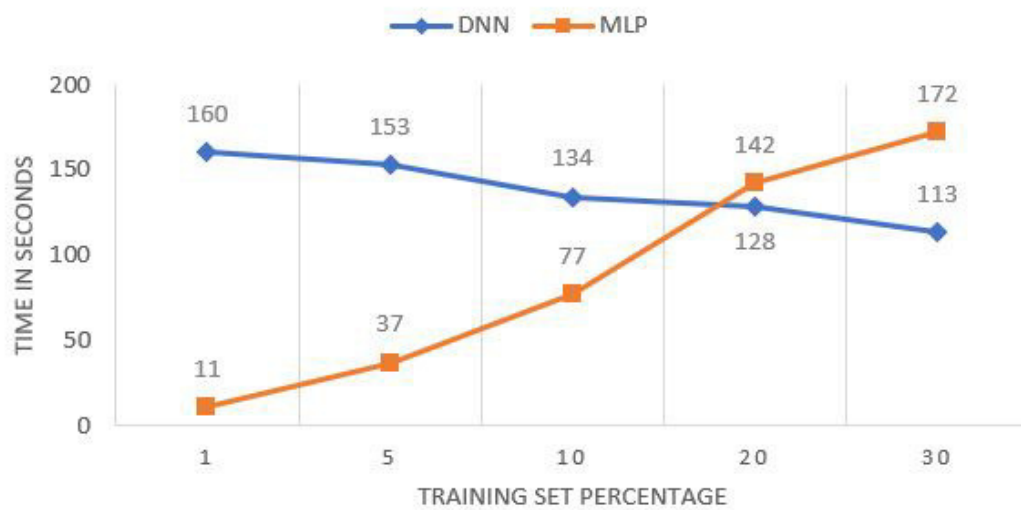


Figure B.1: MLP classifier versus DNN classifier execution time comparison using Sentinel 2 data.

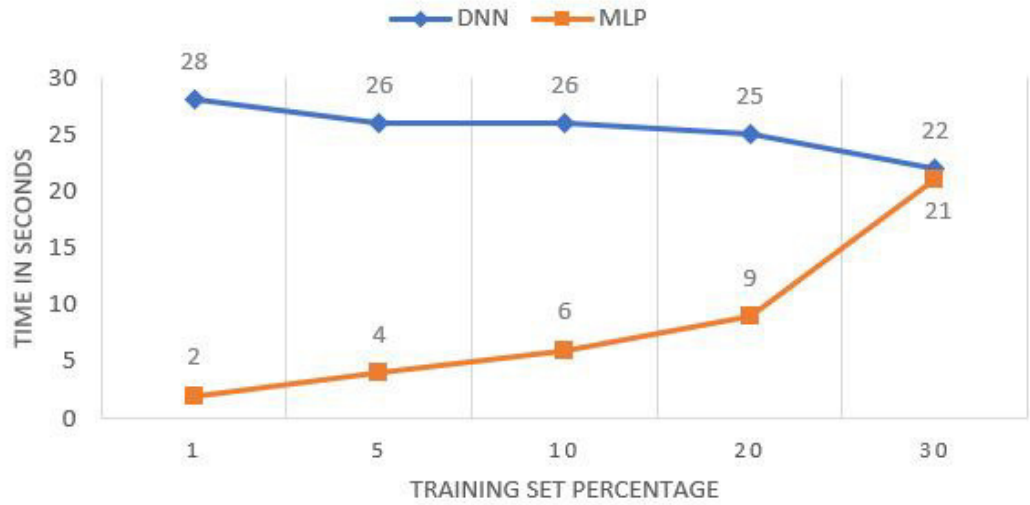


Figure B.2: MLP classifier versus DNN classifier execution time comparison using Landsat 8 data.

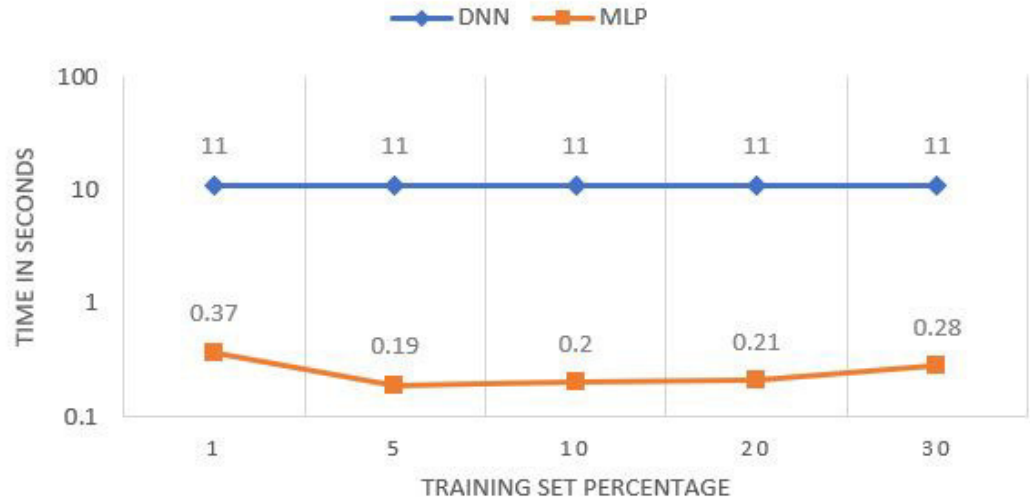


Figure B.3: MLP classifier versus DNN classifier execution time comparison using MODIS data.

B.2 Gradient Boosting



Figure B.4: GB classifier versus XGB classifier execution time comparison using Sentinel 2 data.



Figure B.5: GB classifier versus XGB classifier execution time comparison using Landsat 8 data.



Figure B.6: GB classifier versus XGB classifier execution time comparison using MODIS data.

B.3 Support Vector Machines

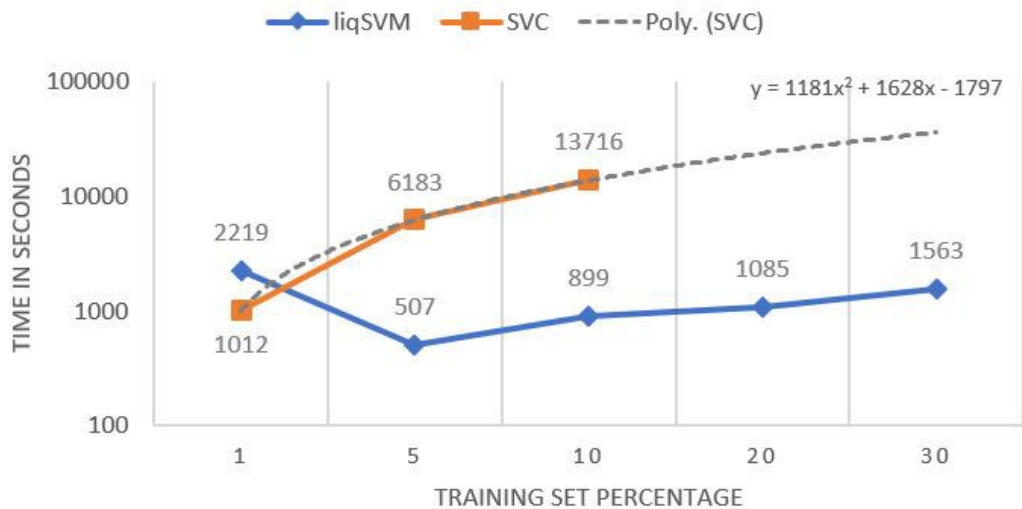


Figure B.7: SVC classifier versus liquidSVM classifier execution time comparison using Sentinel 2 data.



Figure B.8: SVC classifier versus liquidSVM classifier execution time comparison using Landsat 8 data.

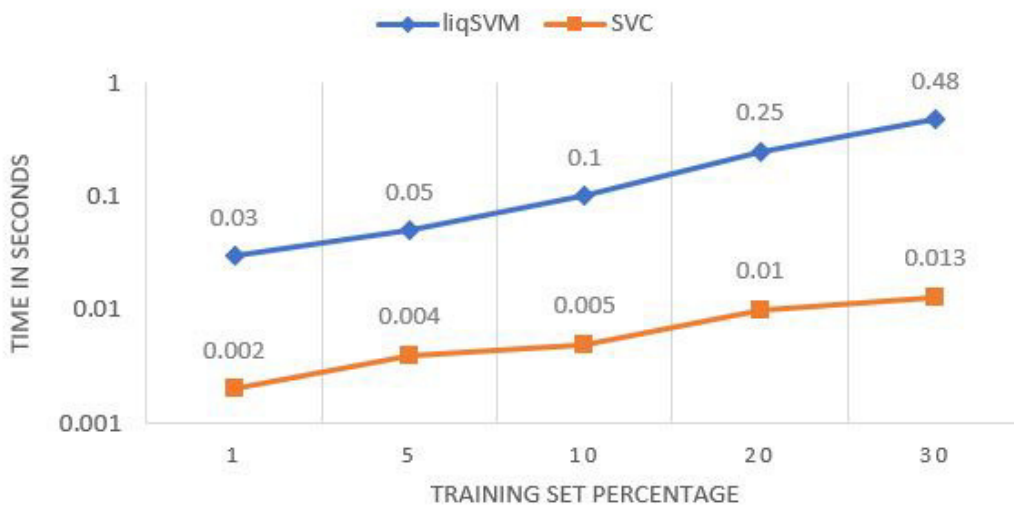


Figure B.9: SVC classifier versus liquidSVM classifier execution time comparison using MODIS data.

CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

C.1 Sentinel 2

C.1.1 Multilayer Perceptron Classifier

Table C.1: Confusion Matrix from the MLP Classifier using 1% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4366733	168428	4535161	0.96
Burned	102423	899599	1002022	0.90
Reference Total	4469156	1068027	Overall Accuracy	0.95
Producer Accuracy	0.98	0.84	Overall Kappa	0.84

Table C.2: Confusion Matrix from the MLP Classifier using 5% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4195181	146464	4341645	0.97
Burned	93404	878410	971814	0.90
Reference Total	4288585	1024874	Overall Accuracy	0.95
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.3: Confusion Matrix from the MLP Classifier using 10% training data from Sentinel 2 imagery

Predicted class	Reference class		Predicted Total	Producer Accuracy
	Not Burned	Burned		
Not Burned	3979586	137911	4117497	0.97
Burned	83283	833023	916306	0.91
Reference Total	4062869	970934	Overall Accuracy	0.96
User Accuracy	0.98	0.86	Overall Kappa	0.86

Table C.4: Confusion Matrix from the MLP Classifier using 20% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3527199	110603	3637802	0.97
Burned	84241	752449	836690	0.90
Reference Total	3611440	863052	Overall Accuracy	0.96
Producer Accuracy	0.98	0.87	Overall Kappa	0.86

Table C.5: Confusion Matrix from the MLP Classifier using 30% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3090080	101760	3191840	0.97
Burned	69929	653411	723340	0.90
Reference Total	3160009	755171	Overall Accuracy	0.96
Producer Accuracy	0.98	0.87	Overall Kappa	0.86

C.1.2 Deep Neural Network Classifier

Table C.6: Confusion Matrix from the DNN Classifier using 1% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4379956	215598	4595554	0.95
Burned	89200	852429	941629	0.91
Reference Total	4469156	1068027	Overall Accuracy	0.94
Producer Accuracy	0.98	0.80	Overall Kappa	0.81

Table C.7: Confusion Matrix from the DNN Classifier using 5% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4190809	191087	4381896	0.96
Burned	97776	833787	931563	0.90
Reference Total	4288585	1024874	Overall Accuracy	0.95
Producer Accuracy	0.98	0.81	Overall Kappa	0.82

Table C.8: Confusion Matrix from the DNN Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3986204	204202	4190406	0.95
Burned	76665	766732	843397	0.91
Reference Total	4062869	970934	Overall Accuracy	0.94
Producer Accuracy	0.98	0.8	Overall Kappa	0.81

Table C.9: Confusion Matrix from the DNN Classifier using 20% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3543274	179722	3722996	0.95
Burned	68166	683330	751496	0.91
Reference Total	3611440	863052	Overall Accuracy	0.94
Producer Accuracy	0.98	0.79	Overall Kappa	0.81

Table C.10: Confusion Matrix from the DNN Classifier using 30% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3094174	155551	3249725	0.95
Burned	65835	599620	665455	0.91
Reference Total	3160009	755171	Overall Accuracy	0.94
Producer Accuracy	0.98	0.8	Overall Kappa	0.82

C.1.3 Gradient Boosting Classifier

Table C.11: Confusion Matrix from the GB Classifier using 1% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4370957	155164	4526121	0.97
Burned	98199	912863	1011062	0.90
Reference Total	4469156	1068027	Overall Accuracy	0.95
Producer Accuracy	0.98	0.85	Overall Kappa	0.85

Table C.12: Confusion Matrix from the GB Classifier using 5% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4196950	143201	4340151	0.97
Burned	91635	881673	973308	0.91
Reference Total	4288585	1024874	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

Table C.13: Confusion Matrix from the GB Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3974394	133914	4108308	0.97
Burned	88475	837020	925495	0.90
Reference Total	4062869	970934	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.86

Table C.14: Confusion Matrix from the GB Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3974394	133914	4108308	0.97
Burned	88475	837020	925495	0.90
Reference Total	4062869	970934	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.86

Table C.15: Confusion Matrix from the GB Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3091898	104620	3196518	0.97
Burned	68111	650551	718662	0.91
Reference Total	3160009	755171	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.86

C.1.4 XGBoost Classifier

Table C.16: Confusion Matrix from the XGB Classifier using 1% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4371621	151620	4523241	0.97
Burned	97535	916407	1013942	0.90
Reference Total	4469156	1068027	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

Table C.17: Confusion Matrix from the XGB Classifier using 5% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4195906	143171	4339077	0.97
Burned	92679	881703	974382	0.90
Reference Total	4288585	1024874	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

Table C.18: Confusion Matrix from the XGB Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3975301	134450	4109751	0.97
Burned	87568	836484	924052	0.91
Reference Total	4062869	970934	5033803	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.86

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.19: Confusion Matrix from the XGB Classifier using 10% training data from Sentinel 2 imagery

	Reference Class			
Predicted Class	Not Burned	Burned	Predicted Total	User Accuracy
Not Burned	3533336	119667	3653003	0.97
Burned	78104	743385	821489	0.90
Reference Total	3611440	863052	4474492	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.86

Table C.20: Confusion Matrix from the XGB Classifier using 10% training data from Sentinel 2 imagery

	Reference Class			
Predicted Class	Not Burned	Burned	Predicted Total	User Accuracy
Not Burned	3091159	104133	3195292	0.97
Burned	68850	651038	719888	0.90
Reference Total	3160009	755171	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.86

C.1.5 Support Vector Classifier

Table C.21: Confusion Matrix from the SVC Classifier using 1% training data from Sentinel 2 imagery

	Reference Class			
Predicted Class	Not Burned	Burned	Predicted Total	User Accuracy
Not Burned	4382393	178314	4560707	0.96
Burned	86763	889713	976476	0.91
Reference Total	4469156	1068027	Overall Accuracy	0.95
Producer Accuracy	0.98	0.83	Overall Kappa	0.84

Table C.22: Confusion Matrix from the SVC Classifier using 5% training data from Sentinel 2 imagery

	Reference Class			
Predicted Class	Not Burned	Burned	Predicted Total	User Accuracy
Not Burned	4203848	162079	4365927	0.96
Burned	84737	862795	947532	0.91
Reference Total	4288585	1024874	Overall Accuracy	0.95
Producer Accuracy	0.98	0.84	Overall Kappa	0.85

Table C.23: Confusion Matrix from the SVC Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3983041	149898	4132939	0.96
Burned	79828	821036	900864	0.91
Reference Total	4062869	970934	Overall Accuracy	0.95
Producer Accuracy	0.98	0.85	Overall Kappa	0.85

C.1.6 liquidSVM Classifier

Table C.24: Confusion Matrix from the liquidSVM Classifier using 1% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4368133	152495	4520628	0.97
Burned	101023	915532	1016555	0.90
Reference Total	4469156	1068027	Overall Accuracy	0.95
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

Table C.25: Confusion Matrix from the liquidSVM Classifier using 5% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4194105	140738	4334843	0.97
Burned	94480	884136	978616	0.90
Reference Total	4288585	1024874	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

Table C.26: Confusion Matrix from the liquidSVM Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3974205	132740	4106945	0.97
Burned	88664	838194	926858	0.90
Reference Total	4062869	970934	Overall Accuracy	0.96
Producer Accuracy	0.98	0.86	Overall Kappa	0.86

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.27: Confusion Matrix from the liquidSVM Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3532461	115188	3647649	0.97
Burned	78979	747864	826843	0.90
Reference Total	3611440	863052	Overall Accuracy	0.96
Producer Accuracy	0.98	0.87	Overall Kappa	0.86

Table C.28: Confusion Matrix from the liquidSVM Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3090441	99230	3189671	0.97
Burned	69568	655941	725509	0.90
Reference Total	3160009	755171	Overall Accuracy	0.96
Producer Accuracy	0.98	0.87	Overall Kappa	0.86

C.1.7 K Nearest Neighbours Classifier

Table C.29: Confusion Matrix from the KNN Classifier using 1% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4364561	173440	4538001	0.96
Burned	104595	894587	999182	0.90
Reference Total	4469156	1068027	Overall Accuracy	0.95
Producer Accuracy	0.98	0.84	Overall Kappa	0.84

Table C.30: Confusion Matrix from the KNN Classifier using 5% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	4188823	154671	4343494	0.96
Burned	99762	870203	969965	0.90
Reference Total	4288585	1024874	Overall Accuracy	0.95
Producer Accuracy	0.98	0.85	Overall Kappa	0.84

Table C.31: Confusion Matrix from the KNN Classifier using 10% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3970301	144358	4114659	0.96
Burned	92568	826576	919144	0.90
Reference Total	4062869	970934	Overall Accuracy	0.95
Producer Accuracy	0.98	0.85	Overall Kappa	0.85

Table C.32: Confusion Matrix from the KNN Classifier using 20% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3528068	125086	3653154	0.97
Burned	83372	737966	821338	0.90
Reference Total	3611440	863052	Overall Accuracy	0.95
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

Table C.33: Confusion Matrix from the KNN Classifier using 30% training data from Sentinel 2 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	3086217	107426	3193643	0.97
Burned	73792	647745	721537	0.90
Reference Total	3160009	755171	Overall Accuracy	0.95
Producer Accuracy	0.98	0.86	Overall Kappa	0.85

C.2 Landsat 8

C.2.1 Multilayer Perceptron Classifier

Table C.34: Confusion Matrix from the MLP Classifier using 1% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	486362	29831	516193	0.94
Burned	10624	88950	99574	0.89
Reference Total	496986	118781	Overall Accuracy	0.93
Producer Accuracy	0.98	0.75	Overall Kappa	0.78

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.35: Confusion Matrix from the MLP Classifier using 5% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	465927	21503	487430	0.96
Burned	10978	92479	103457	0.89
Reference Total	476905	113982	Overall Accuracy	0.95
Producer Accuracy	0.98	0.81	Overall Kappa	0.82

Table C.36: Confusion Matrix from the MLP Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	437598	15691	453289	0.97
Burned	14207	92292	106499	0.87
Reference Total	451805	107983	Overall Accuracy	0.95
Producer Accuracy	0.97	0.85	Overall Kappa	0.82

Table C.37: Confusion Matrix from the MLP Classifier using 20% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	390923	15427	406350	0.96
Burned	10681	80558	91239	0.88
Reference Total	401604	95985	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.38: Confusion Matrix from the MLP Classifier using 30% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	340247	11568	351815	0.97
Burned	11157	72419	83576	0.87
Reference Total	351404	83987	Overall Accuracy	0.95
Producer Accuracy	0.97	0.86	Overall Kappa	0.83

C.2.2 Deep Neural Network Classifier

Table C.39: Confusion Matrix from the DNN Classifier using 1% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	483543	25262	508805	0.95
Burned	13443	93519	106962	0.87
Reference Total	496986	118781	Overall Accuracy	0.94
Producer Accuracy	0.97	0.79	Overall Kappa	0.78

Table C.40: Confusion Matrix from the DNN Classifier using 5% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	463847	24189	488036	0.95
Burned	13058	89793	102851	0.87
Reference Total	476905	113982	Overall Accuracy	0.94
Producer Accuracy	0.97	0.79	Overall Kappa	0.80

Table C.41: Confusion Matrix from the DNN Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	439806	24950	464756	0.95
Burned	11999	83033	95032	0.87
Reference Total	451805	107983	Overall Accuracy	0.93
Producer Accuracy	0.97	0.77	Overall Kappa	0.78

Table C.42: Confusion Matrix from the DNN Classifier using 20% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	439806	24950	464756	0.95
Burned	11999	83033	95032	0.87
Reference Total	451805	107983	Overall Accuracy	0.93
Producer Accuracy	0.97	0.77	Overall Kappa	0.78

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.43: Confusion Matrix from the DNN Classifier using 30% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	342152	17218	359370	0.95
Burned	9252	66769	76021	0.88
Reference Total	351404	83987	Overall Accuracy	0.94
Producer Accuracy	0.97	0.79	Overall Kappa	0.80

C.2.3 Gradient Boosting Classifier

Table C.44: Confusion Matrix from the GBC Classifier using 1% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	482350	19832	502182	0.96
Burned	14636	98949	113585	0.87
Reference Total	496986	118781	Overall Accuracy	0.94
Producer Accuracy	0.97	0.83	Overall Kappa	0.82

Table C.45: Confusion Matrix from the GBC Classifier using 5% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	463577	17688	481265	0.96
Burned	13328	96294	109622	0.88
Reference Total	476905	113982	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.46: Confusion Matrix from the GBC Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	439453	17046	456499	0.96
Burned	12352	90937	103289	0.88
Reference Total	451805	107983	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.47: Confusion Matrix from the GBC Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	391108	15024	406132	0.96
Burned	10496	80961	91457	0.89
Reference Total	401604	95985	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.48: Confusion Matrix from the GBC Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	342169	12984	355153	0.96
Burned	9235	71003	80238	0.88
Reference Total	351404	83987	Overall Accuracy	0.95
Producer Accuracy	0.97	0.85	Overall Kappa	0.83

C.2.4 XGBoost Classifier

Table C.49: Confusion Matrix from the XGB Classifier using 1% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	484231	19907	504138	0.96
Burned	12755	98874	111629	0.89
Reference Total	496986	118781	Overall Accuracy	0.95
Producer Accuracy	0.97	0.83	Overall Kappa	0.83

Table C.50: Confusion Matrix from the XGB Classifier using 5% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	463924	17513	481437	0.96
Burned	12981	96469	109450	0.88
Reference Total	476905	113982	Overall Accuracy	0.95
Producer Accuracy	0.97	0.85	Overall Kappa	0.83

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.51: Confusion Matrix from the XGB Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	439949	16863	456812	0.96
Burned	11856	91120	102976	0.88
Reference Total	451805	107983	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.52: Confusion Matrix from the XGB Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	391118	14888	406006	0.96
Burned	10486	81097	91583	0.89
Reference Total	401604	95985	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.53: Confusion Matrix from the XGB Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	342373	13176	355549	0.96
Burned	9031	70811	79842	0.89
Reference Total	351404	83987	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

C.2.5 Support Vector Classifier

Table C.54: Confusion Matrix from the SVC Classifier using 1% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	484881	21760	506641	0.96
Burned	12105	97021	109126	0.89
Reference Total	496986	118781	Overall Accuracy	0.95
Producer Accuracy	0.98	0.82	Overall Kappa	0.83

Table C.55: Confusion Matrix from the SVC Classifier using 5% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	465874	20697	486571	0.96
Burned	11031	93285	104316	0.89
Reference Total	476905	113982	Overall Accuracy	0.95
Producer Accuracy	0.98	0.82	Overall Kappa	0.83

Table C.56: Confusion Matrix from the SVC Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	441052	19049	460101	0.96
Burned	10753	88934	99687	0.89
Reference Total	451805	107983	Overall Accuracy	0.95
Producer Accuracy	0.98	0.82	Overall Kappa	0.83

Table C.57: Confusion Matrix from the SVC Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	392002	16618	408620	0.96
Burned	9602	79367	88969	0.89
Reference Total	401604	95985	Overall Accuracy	0.95
Producer Accuracy	0.98	0.83	Overall Kappa	0.84

Table C.58: Confusion Matrix from the SVC Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	343047	14541	357588	0.96
Burned	8357	69446	77803	0.89
Reference Total	351404	83987	Overall Accuracy	0.95
Producer Accuracy	0.98	0.83	Overall Kappa	0.84

C.2.6 liquidSVM Classifier

Table C.59: Confusion Matrix from the liquidSVM Classifier using 1% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	483742	19582	503324	0.96
Burned	13244	99199	112443	0.88
Reference Total	496986	118781	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.60: Confusion Matrix from the liquidSVM Classifier using 5% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	463870	17788	481658	0.96
Burned	13035	96194	109229	0.88
Reference Total	476905	113982	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

Table C.61: Confusion Matrix from the liquidSVM Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	439624	16454	456078	0.96
Burned	12181	91529	103710	0.88
Reference Total	451805	107983	Overall Accuracy	0.95
Producer Accuracy	0.97	0.85	Overall Kappa	0.83

Table C.62: Confusion Matrix from the liquidSVM Classifier using 20% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	390791	14176	404967	0.96
Burned	10813	81809	92622	0.88
Reference Total	401604	95985	Overall Accuracy	0.95
Producer Accuracy	0.97	0.85	Overall Kappa	0.84

Table C.63: Confusion Matrix from the liquidSVM Classifier using 30% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	342136	12578	354714	0.96
Burned	9268	71409	80677	0.89
Reference Total	351404	83987	Overall Accuracy	0.95
Producer Accuracy	0.97	0.85	Overall Kappa	0.84

C.2.7 K Nearest Neighbours Classifier

Table C.64: Confusion Matrix from the KNN Classifier using 1% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	483112	22124	505236	0.96
Burned	13874	96657	110531	0.87
Reference Total	496986	118781	Overall Accuracy	0.94
Producer Accuracy	0.97	0.81	Overall Kappa	0.81

Table C.65: Confusion Matrix from the KNN Classifier using 5% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	463347	19473	482820	0.96
Burned	13558	94509	108067	0.87
Reference Total	476905	113982	Overall Accuracy	0.94
Producer Accuracy	0.97	0.83	Overall Kappa	0.82

Table C.66: Confusion Matrix from the KNN Classifier using 10% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	439105	18039	457144	0.96
Burned	12700	89944	102644	0.88
Reference Total	451805	107983	Overall Accuracy	0.95
Producer Accuracy	0.97	0.83	Overall Kappa	0.82

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.67: Confusion Matrix from the KNN Classifier using 20% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	390345	15456	405801	0.96
Burned	11259	80529	91788	0.88
Reference Total	401604	95985	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.82

Table C.68: Confusion Matrix from the KNN Classifier using 30% training data from Landsat 8 imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	341611	13471	355082	0.96
Burned	9793	70516	80309	0.88
Reference Total	351404	83987	Overall Accuracy	0.95
Producer Accuracy	0.97	0.84	Overall Kappa	0.83

C.3 MODIS

C.3.1 Multilayer Perceptron Classifier

Table C.69: Confusion Matrix from the MLP Classifier using 1% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1466	95	1561	0.94
Burned	161	288	449	0.64
Reference Total	1627	383	Overall Accuracy	0.87
Producer Accuracy	0.90	0.75	Overall Kappa	0.61

Table C.70: Confusion Matrix from the MLP Classifier using 5% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1473	135	1608	0.92
Burned	88	233	321	0.73
Reference Total	1561	368	Overall Accuracy	0.88
Producer Accuracy	0.94	0.63	Overall Kappa	0.60

Table C.71: Confusion Matrix from the MLP Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1399	139	1538	0.91
Burned	80	209	289	0.72
Reference Total	1479	348	Overall Accuracy	0.88
Producer Accuracy	0.95	0.60	Overall Kappa	0.59

Table C.72: Confusion Matrix from the MLP Classifier using 20% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1283	183	1466	0.88
Burned	31	127	158	0.80
Reference Total	1314	310	Overall Accuracy	0.87
Producer Accuracy	0.98	0.41	Overall Kappa	0.5

Table C.73: Confusion Matrix from the MLP Classifier using 30% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1097	107	1204	0.91
Burned	53	164	217	0.76
Reference Total	1150	271	Overall Accuracy	0.89
Producer Accuracy	0.95	0.61	Overall Kappa	0.61

C.3.2 Deep Neural Network Classifier

Table C.74: Confusion Matrix from the DNN Classifier using 1% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1542	213	1755	0.88
Burned	85	170	255	0.67
Reference Total	1627	383	Overall Accuracy	0.85
Producer Accuracy	0.95	0.44	Overall Kappa	0.56

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.75: Confusion Matrix from the DNN Classifier using 5% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1433	100	1533	0.93
Burned	128	268	396	0.68
Reference Total	1561	368	Overall Accuracy	0.88
Producer Accuracy	0.92	0.73	Overall Kappa	0.57

Table C.76: Confusion Matrix from the DNN Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1438	185	1623	0.89
Burned	41	163	204	0.80
Reference Total	1479	348	Overall Accuracy	0.88
Producer Accuracy	0.97	0.47	Overall Kappa	0.52

Table C.77: Confusion Matrix from the DNN Classifier using 20% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1258	130	1388	0.91
Burned	56	180	236	0.76
Reference Total	1314	310	Overall Accuracy	0.89
Producer Accuracy	0.96	0.58	Overall Kappa	0.61

Table C.78: Confusion Matrix from the DNN Classifier using 30% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1070	90	1160	0.92
Burned	80	181	261	0.69
Reference Total	1150	271	Overall Accuracy	0.88
Producer Accuracy	0.93	0.67	Overall Kappa	0.61

C.3.3 Gradient Boosting Classifier

Table C.79: Confusion Matrix from the GBC Classifier using 1% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1516	191	1707	0.89
Burned	111	192	303	0.63
Reference Total	1627	383	Overall Accuracy	0.85
Producer Accuracy	0.93	0.50	Overall Kappa	0.42

Table C.80: Confusion Matrix from the GBC Classifier using 5% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1475	162	1637	0.90
Burned	86	206	292	0.71
Reference Total	1561	368	Overall Accuracy	0.87
Producer Accuracy	0.94	0.56	Overall Kappa	0.48

Table C.81: Confusion Matrix from the GBC Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1411	178	1589	0.89
Burned	68	170	238	0.71
Reference Total	1479	348	Overall Accuracy	0.87
Producer Accuracy	0.95	0.49	Overall Kappa	0.54

Table C.82: Confusion Matrix from the GBC Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1248	127	1375	0.91
Burned	66	183	249	0.73
Reference Total	1314	310	Overall Accuracy	0.88
Producer Accuracy	0.95	0.59	Overall Kappa	0.58

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.83: Confusion Matrix from the GBC Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1085	102	1187	0.91
Burned	65	169	234	0.72
Reference Total	1150	271	Overall Accuracy	0.88
Producer Accuracy	0.94	0.62	Overall Kappa	0.58

C.3.4 XGBoost Classifier

Table C.84: Confusion Matrix from the XGB Classifier using 5% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1459	142	1601	0.91
Burned	102	226	328	0.69
Reference Total	1561	368	Overall Accuracy	0.87
Producer Accuracy	0.93	0.61	Overall Kappa	0.58

Table C.85: Confusion Matrix from the XGB Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1394	156	1550	0.90
Burned	85	192	277	0.69
Reference Total	1479	348	Overall Accuracy	0.87
Producer Accuracy	0.94	0.55	Overall Kappa	0.57

Table C.86: Confusion Matrix from the XGB Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1242	111	1353	0.92
Burned	72	199	271	0.73
Reference Total	1314	310	Overall Accuracy	0.89
Producer Accuracy	0.95	0.64	Overall Kappa	0.62

Table C.87: Confusion Matrix from the XGB Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1060	99	1159	0.91
Burned	90	172	262	0.66
Reference Total	1150	271	Overall Accuracy	0.87
Producer Accuracy	0.92	0.63	Overall Kappa	0.62

C.3.5 Support Vector Classifier

Table C.88: Confusion Matrix from the SVC Classifier using 1% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1511	124	1635	0.92
Burned	116	259	375	0.69
Reference Total	1627	383	Overall Accuracy	0.88
Producer Accuracy	0.93	0.68	Overall Kappa	0.61

Table C.89: Confusion Matrix from the SVC Classifier using 5% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1514	183	1697	0.89
Burned	47	185	232	0.80
Reference Total	1561	368	Overall Accuracy	0.88
Producer Accuracy	0.97	0.50	Overall Kappa	0.61

Table C.90: Confusion Matrix from the SVC Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1437	166	1603	0.90
Burned	42	182	224	0.81
Reference Total	1479	348	Overall Accuracy	0.89
Producer Accuracy	0.97	0.52	Overall Kappa	0.61

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.91: Confusion Matrix from the SVC Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1273	144	1417	0.90
Burned	41	166	207	0.80
Reference Total	1314	310	Overall Accuracy	0.89
Producer Accuracy	0.97	0.54	Overall Kappa	0.59

Table C.92: Confusion Matrix from the SVC Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1109	121	1230	0.90
Burned	41	150	191	0.79
Reference Total	1150	271	Overall Accuracy	0.89
Producer Accuracy	0.96	0.55	Overall Kappa	0.60

C.3.6 liquidSVM Classifier

Table C.93: Confusion Matrix from the liquidSVM Classifier using 1% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1599	239	1838	0.87
Burned	28	144	172	0.84
Reference Total	1627	383	Overall Accuracy	0.87
Producer Accuracy	0.98	0.38	Overall Kappa	0.45

Table C.94: Confusion Matrix from the liquidSVM Classifier using 5% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1529	210	1739	0.88
Burned	32	158	190	0.83
Reference Total	1561	368	Overall Accuracy	0.87
Producer Accuracy	0.98	0.43	Overall Kappa	0.50

Table C.95: Confusion Matrix from the liquidSVM Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1409	160	1569	0.90
Burned	70	188	258	0.73
Reference Total	1479	348	Overall Accuracy	0.87
Producer Accuracy	0.95	0.54	Overall Kappa	0.55

Table C.96: Confusion Matrix from the liquidSVM Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1277	158	1435	0.89
Burned	37	152	189	0.80
Reference Total	1314	310	Overall Accuracy	0.88
Producer Accuracy	0.97	0.49	Overall Kappa	0.54

Table C.97: Confusion Matrix from the liquidSVM Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1099	103	1202	0.91
Burned	51	168	219	0.77
Reference Total	1150	271	Overall Accuracy	0.89
Producer Accuracy	0.96	0.62	Overall Kappa	0.62

C.3.7 K Nearest Neighbours Classifier

Table C.98: Confusion Matrix from the KNN Classifier using 1% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1584	242	1826	0.87
Burned	43	141	184	0.77
Reference Total	1627	383	Overall Accuracy	0.86
Producer Accuracy	0.97	0.37	Overall Kappa	0.45

APPENDIX C. CONFUSION MATRICES FOR BURNED AREA CLASSIFICATION

Table C.99: Confusion Matrix from the KNN Classifier using 5% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1456	131	1587	0.92
Burned	105	237	342	0.69
Reference Total	1561	368	Overall Accuracy	0.88
Producer Accuracy	0.93	0.64	Overall Kappa	0.54

Table C.100: Confusion Matrix from the KNN Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1365	124	1489	0.92
Burned	114	224	338	0.66
Reference Total	1479	348	Overall Accuracy	0.87
Producer Accuracy	0.92	0.64	Overall Kappa	0.57

Table C.101: Confusion Matrix from the KNN Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1212	109	1321	0.92
Burned	102	201	303	0.66
Reference Total	1314	310	Overall Accuracy	0.87
Producer Accuracy	0.92	0.65	Overall Kappa	0.58

Table C.102: Confusion Matrix from the KNN Classifier using 10% training data from MODIS imagery

Predicted Class	Reference Class		Predicted Total	User Accuracy
	Not Burned	Burned		
Not Burned	1074	98	1172	0.92
Burned	76	173	249	0.69
Reference Total	1150	271	Overall Accuracy	0.88
Producer Accuracy	0.93	0.64	Overall Kappa	0.59