

Technical Efficiency of Wildfire Detection and Machine Learning Predictions in Alberta, Canada

by

Vaibhav Manawat

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Agricultural and Resource Economics

Department of Resource Economics and Environmental Sociology

University of Alberta

© Vaibhav Manawat, 2023

Abstract

Wildfire management agencies must continue to evolve and adjust to the dynamic nature of their industry. They face pressures that include a changing climate with the prospect of intense future fire seasons, tighter government budgets for wildfire detection and suppression, and the fast-pace of development of wildfire detection technologies such as remote-sensing devices, satellites, and drones. Alberta Wildfire, to fulfil its mandate of managing wildfire in the Forest Protection Area of Alberta, must adapt to these conditions by making cost-effective strategic decisions. As such, there is an increasing need for studies that examine the performance of Alberta's wildfire detection system. This study offers two main contributions. First, we provide insights on the contribution of lookout towers in detecting wildfires and their role in the entire detection system. A production economics approach is employed to develop robust non-parametric Data Envelopment Analysis (DEA) models that estimate production frontiers that serve as a technical benchmark for lookout towers in Alberta's detection system. Results from this analysis reveal a high-performing detection system and most lookout towers have high technical efficiency. Lookout towers operate close to the "state-of-the-art" technology frontier such that further productivity gains may require a new technology. Second, we explore the relationship between technical efficiency of lookout towers and local weather. We develop machine learning models that use local weather to classify lookout towers as technically (or productively) efficient or not. We find that weather can successfully predict a tower's technical efficiency class, which suggests that there is a strong association between local weather and technical efficiency of wildfire detection.

Preface

This thesis is an original work by Vaibhav Manawat under the supervision of Dr. Bruno Wichmann. No part of this thesis has been previously published. There was no experiment involving test subjects, so there was no need for an ethics approval. The project was funded by Alberta Forestry and Parks' Wildfire Management Branch, Alberta Wildfire, through Canada Wildfire.

Acknowledgments

I would like to extend my appreciation to Alberta Wildfire for providing funding and data to make this project possible. Special recognition goes to Dave Schroeder and Michael Kakoullis for their continuous help in explaining wildfire detection concepts, gathering data, discussing the scope of the project, and giving me the opportunity to present preliminary findings at the 5th National Wildfire Detection Workshop. Their expertise and contributions were crucial for the success of this research study.

I thank my supervisor, Bruno Wichmann. His guidance at every stage of my research encouraged me to do my best. I appreciate his contributions very much. He gave me all the time I could ask for and his mentorship imparted essential technical and professional skills. I am truly grateful.

I would like to thank the members of my thesis defense committee. Thank you to Dr. Xiaoli Fan and Dr. Jen Beverly for valuable comments and suggestions. The points raised will help me improve my future work and contribute to my growth. Additional thanks go to Dr. Philippe Marcoul for chairing my defense in a courteous and equitable manner.

Thank you to other professors in REES who consistently fostered a healthy learning environment. I would like to mention that my classmates and students from the year above were very helpful throughout my degree.

I am very thankful to my parents, Lakshmi, and Nitin, for their unconditional support in my decision to pursue this degree. I would like to thank my brother, Vedant, for looking at my codes when I needed it the most. Stanley, my quirky cat, I would like to thank you for napping right under my monitors and yelling at me to take a break, even though you maybe just wanted treats.

Table of Contents

Abstract.....	ii
Preface.....	iii
Acknowledgments	iv
Table of Contents	v
List of Tables	vii
List of Figures.....	viii
Chapter 1. Introduction	1
1.1. Preview of Main Results.....	6
1.2. Scope of the Work	7
1.3. Thesis Structure	9
Chapter 2. Data	10
Chapter 3. Lookout Towers and Wildfire Detection	13
Chapter 4. Technical Efficiency Framework	20
4.1. Technical Efficiency	20
4.2. Parametric vs Nonparametric Estimates of Technical Efficiency	23
4.3. Bias-corrected DEA Scores	24
4.4. Wildfire Detection Technical Efficiency Model.....	32
Chapter 5. Technical Efficiency Estimates for Wildfire Detection	40
Chapter 6. Machine Learning to Predict Technical Efficiency	47
6.1. Weather Data as Machine Learning Predictors	50
6.2. Machine Learning Algorithms.....	53
6.3 Model Training and Hyperparameters	59
6.4. Model Assessment	62
6.5. Feature Importance	69
Chapter 7. Discussion	75

7.1. Limitations	79
Bibliography	82
Appendix A: Forest Protection Area in Alberta	93
Appendix B: Fires outside Surveillance Regions	94
Appendix C: Output-oriented DEA Model (Variable Returns to Scale).....	95
Appendix D: Types of Bootstrapping.....	97
Appendix E: Yearly Distribution of Technical Efficiency Scores	98
Appendix F: Effect of outliers in Data Envelopment Analysis	101
Appendix G: Map of Weather Stations	104
Appendix H: ACIS' Estimation for Modelling Incoming Radiation	105
Appendix I: Machine Learning Model Specific Hyperparameters.....	107
Appendix J: Performance Measures for Machine Learning Models.....	110
Appendix K: Feature Importance	113
Appendix L: Robustness Check for AUROCs and Feature Importance.....	115
Appendix M: Short-run versus Long-run Production Functions	118
Appendix N: Code Availability and Replication Guide for OSF	122

List of Tables

Table 2.1 Summary Statistics of Fire-level Variables	11
Table 2.2 Summary Statistics of Tower-level Variables	11
Table 2.3 Summary Statistics of Weather Variables	12
Table 3.1 Detection Frequencies of Different Detection Agents over Time	15
Table 4.1 Input and Output Variables for Data Envelopment Analysis	38
Table 4.2 DEA Models for Technical Efficiency Analysis	39
Table 5.1 Bias-corrected Technical Efficiency Scores Pooled across All Years	40
Table 5.2 Proportion of Technically Efficient Lookout Towers Based on Different Thresholds	42
Table 5.3 Regression Results.....	44
Table 6.1 Weather Variables as Predictors in Machine Learning Models	52
Table I.1 Model-Specific Hyperparameters.....	107
Table J.1 Output Metrics for ML Models.....	112
Table K.1 Model-specific Permutation Feature Importance.....	113
Table K.2 AUROC Weighted Feature Importance.....	113
Table L.1 AUROC Values for Different Classification Criteria	115
Table L.2 Performance Categories for Classifying AUROC Values	116
Table L.3 AUROC Performance Categories for Different Classification Criteria.....	116
Table L.4 Feature Importance across Different Classification Criteria	117

List of Figures

Figure 3.1 Fire Locations and Lookout Tower Surveillance Regions in Alberta.....	13
Figure 3.2 Frequency of Fires by Size Class	15
Figure 3.3 Number of Fires and Contribution of Lookout Towers over Time	17
Figure 3.4 Frequency of Detections outside Surveillance Region.....	19
Figure 4.1 Production Possibility Set and Distance Functions	21
Figure 4.2 Bias in Constructing DEA Frontier	26
Figure 4.3 Theoretical Frontier vs DEA Frontier vs Bootstrap Frontier	29
Figure 4.4 Proportion of Fires Detected by Lookout Towers	33
Figure 4.5 Average Early Detection Times Over Time	35
Figure 4.6 Line of Sight Viewshed Analysis for AB Wildfire Lookout Towers.....	36
Figure 4.7 Lookout Tower Visibility Map.....	37
Figure 5.1 Bias-corrected Technical Efficiency Scores Over Time	41
Figure 5.2 Classification of Lookout Towers	43
Figure 5.3 Heat Map of Average Technical Efficiency Scores	46
Figure 6.1 Confusion Matrix.....	63
Figure 6.2 General ROC	65
Figure 6.3 AUROC Graphs for Machine Learning Models	68
Figure 6.4 Permutation Feature Importance	71
Figure 6.5 AUROC Weighted Permutation Feature Importance.....	74
Figure A.1 Forest Protection Areas of Alberta	93
Figure B.1 Frequency of Wildfires outside Surveillance Regions, by Size Class and Year	94
Figure E.1 Histogram of Technical Efficiency Estimates for Model A, 2006 – 2013	99
Figure E.2 Histogram of Technical Efficiency Estimates for Model A, 2014 – 2021	100
Figure F.1 Production Frontier and Outliers in Data	102
Figure G.1 Map of Weather Stations in Alberta	104
Figure K.1 Model Weights and AUROC.....	114
Figure M.1 Short-run Production Function	119
Figure M.2 Long-run Production Function.....	120

Chapter 1. Introduction

Forest fires have been increasing in severity and intensity worldwide (Westerling et al. 2006; Moreira and Pe'er 2018; Fernandes et al. 2016). Wildfires can quickly escape a containable size, cause destruction to the ecosystem, and endanger lives within a matter of minutes. In addition to local damage, wildfires impose negative externalities on a global scale. Wildland fires contributed to a substantial 23% of global forest loss in 2018 (Curtis et al. 2018). The wildfires in Canada in 2017 emitted significant amounts of smoke that remained detectable in the stratosphere for eight months (Yu et al. 2019). The Australian wildfires in 2019-2020 produced enough smoke and pollutants to significantly damage the ozone chemistry (Bernath, Boone and Crouse 2022).

Climate change leads to unexpected changes in global fire activity (Flannigan et al. 2009). It contributes to earlier snowmelts that are strongly correlated to higher temperatures during spring and summer months. This crucial time period marks the start of fire seasons (Westerling et al. 2006). Higher temperatures provide ideal conditions for wildfires, and as a result wildfires are expected to increase in both duration and frequency, further extending the severity of fire seasons (Walker et al. 2019; Halofsky, Peterson and Harvey 2020; Hanes et al. 2019). Thus, considering these growing concerns, detecting wildfires early is increasingly important as future fire seasons and average temperatures are estimated to be more uncertain.

Detecting wildfires is a crucial step in mitigating damage. In some cases, fires that go undetected in their initial phase may become destructive and costly wildfires. These fires increase the burden on management programs as they may have bigger burned areas and impose higher risks (Mendes, 2010). Early detection can reduce the probability of fires escaping out of control

and provides an opportunity for rapid suppression while minimizing firefighting costs (McFayden et al. 2019; Lindenmayer, Zylstra and Yebra 2022).

Organizations responsible for wildfire detection programs utilize a mix of direct and indirect detection agents. McFayden et al. (2019) describe a wildfire detection system by classifying agents into the two categories. Detection agents such as aerial patrols, lookout towers and ground crews are dispatched at specific times or time periods to actively search for fires and in this sense are considered to be ‘direct’ (or planned) agents. On the other hand, ‘indirect’ (or unplanned) detection is conducted by passive agents that relay information about possible wildfires. These agents consist of the general public (e.g., campers or hikers), unplanned aircrafts (e.g., commercial, or private aircrafts flying over the wildlands and spotting potential fires), private industry workers, forest rangers, and reporting of fires using a hotline (McFayden et al., 2019; MNP LLP, 2020). These direct and indirect detection agents are also referred to as traditional agents as they have played a significant role in detecting wildfires in various parts of the world (Rego et al. 2013).

Lookout towers are the most common planned wildfire detection agent across the world (Rego et al. 2013; Pompa-García et al. 2010; Kucuk et al. 2017; Fantini et al. 2022)¹. These towers are strategically constructed with the goal of maximizing visibility of wildlands. As lookout towers have fixed locations, they provide continuous coverage of forest lands with the potential to immediately report wildfires in unpopulated regions (MNP LLP 2020). Furthermore, lookout towers have the capacity to adapt to new technologies that can supplement their performance. Traditionally, lookout towers are manned by observers that use a set of tools to carry out detection

¹ Kucuk et al., 2017 is based in Turkey. Pompa-Garcia et al., 2010 is based in Northern Mexico. Rego et al., 2013 look at Portugal and Spain. Fantini et al., 2022 is based in Sardinia. This literature shows various developed and developing nations that rely on lookout towers for wildfire detection.

(e.g., compasses, maps, scopes, and the Osborne Fire Finder). Lookout towers can be fitted with new technologies such as high-definition cameras and remote sensing devices to support the performance of observers (Bao et al. 2015). As lookout towers are already strategically placed in the wildfire landscape, the financial burden of implementing new technologies is reduced.

This study focuses on wildfire detection in Alberta, Canada. The detection system developed by Alberta's Wildfire Management Branch, i.e., Alberta Wildfire, offers an opportunity to examine lookout towers and many factors make Alberta an interesting case study. First, Alberta uses multiple commonly used traditional agents to carry out detection: aerial and ground patrols, lookout towers, and a public hotline. Lookout towers are an integral part of the system and have been used in Alberta for almost a century, partially because lookout towers are excellent in areas that are difficult to access by road. Lookout towers also play a major role in Alberta's communication network and have the additional benefit of not obstructing wildlife (MNP LLP 2020)². Second, over the period of this research (2006 – 2021), Alberta Wildfire has been subject to significant fiscal challenges. Budget constraints have forced the agency to reduce the number of lookout towers. Our data indicates that, in 2006, 112 towers were active (i.e. reported fires) while this number decreased to 72 in 2021. Balancing the detection of wildfire in the context of a changing climate with a contracting government budget is a challenging task. Therefore, there is an increasing need for studies to develop methodologies for estimating the productive performance of lookout systems (chapters 4 and 5) and the drivers of technically efficient wildfire detection (chapter 6).

² The lookout tower system helped with Alberta's First Responders Radio Communication System. A two-way radio used by first responders in remote areas.

This thesis offers two main contributions. First, we employ a production economics approach to wildfire detection that considers lookout towers as production units. We conceptualize a production process where the visibility profile of lookout towers serves as productive inputs to produce prompt and accurate wildfire detection. The main statistic estimated is the output-oriented technical efficiency score of lookout towers. Given a production characterization (i.e. technology), and holding the visibility profile of towers constant, the technical efficiency score indicates the proportional (or radial) increase in detection output (if any) that is necessary to place a lookout tower at the technical efficiency frontier.

The approach is reminiscent of the detection function approach, where wildfire detection (extensive margin) is modeled as a function of detection inputs such as distance, topography, and air transparency (Rego and Catry 2006). In our production approach, the production of wildfire detection is framed in a process with two outputs. Specifically, we employ output measures at both the extensive (wildfire is detected or not) and intensive (wildfire detection time, from ignition to detection) margins to characterize production. For inputs, our model assumes that each lookout tower utilizes their visibility profile (range and characterization of visibility, e.g., directly vs indirectly visible area) to survey the land and report locations of potential wildfires.

Technical efficiency scores are estimated using the non-parametric methodology of data envelopment analysis (DEA). No specific functional form is assumed, and the production frontier is determined by the outer envelope of the data. We use a bias-correction approach to correct for finite sample biases of technical efficiency scores based on envelopment techniques (Simar and Wilson 1998). The analysis allows us to let the data drive the construction of the technical efficiency frontier. This frontier is useful for policy as it reports the maximum (or technologically feasible) detection output given the visibility profile of each tower. Since visibility profiles are

fixed in the short-run, the analysis has the potential to inform long-run input adjustments. Considering the short-run variation in outputs, the analysis offers measurements of technical efficiency over multiple fire seasons. As the data envelopment is season-specific, the technical efficiency estimation is robust to technological changes and allows for the characterization of productive efficiency trends.

The second main contribution is to develop insights into the drivers of technical efficiency. While the detection function literature sets out to fulfil a similar task by relying on a detection function (Rego and Catry 2006), we again rely on non-parametric models where the shape of the impact of drivers (e.g., weather) is determined by the data itself therefore avoiding misspecification biases. We develop machine learning models to classify lookout towers as technically efficient (production frontier units) or not technically efficient (units below the production frontier) based on weather variables such as air temperature, relative humidity, incoming radiation, precipitation and snow water equivalent, and wind speed. Machine learning models are trained using a sample of the dataset, their hyperparameters are tuned for optimal predictions, and then the models carry out predictions on unseen data points.

The data for this study comes from two sources and encompasses the period from 2006 to 2021. First, the data for wildfire detection production (outputs and inputs) was provided by AB Wildfire. The detection data includes key information on all wildfires that were used to build outputs (e.g., dates, detection agent and times, size, and location). The data also contains information on each lookout tower's visibility profile, which is essential to measure detection inputs. Second, to examine the role of local weather on technical efficiency, we use data from the Alberta Climate Information Service. The ACIS collects meteorological data using numerous weather stations across the province, including lookout tower weather station observations. We

leverage their dataset for the purpose of predicting technical efficiency as a function of weather around a lookout tower, which allows us to rank the role of various weather components in driving technically efficient wildfire detection.

1.1. Preview of Main Results

Regarding the evaluation of lookout towers' performance, we find the average bias-corrected (output-oriented) technical efficiency score to be 1.059. Holding the visibility profile of towers fixed, lookout towers must increase their detection output by an average of 5.9% to achieve the technical efficiency frontier. Using our main model, approximately 60% of towers are classified as technically efficient. We also find that the detection system becomes more technically efficient over time. In 2006, average bias-corrected (output-oriented) technical efficiency is 1.088 and decreases to 1.024 in 2021.

Regarding the drivers of lookout towers' performance, our approach is to use machine learning algorithms to examine the role of weather variables (features) in predicting lookout tower (binary) technical efficiency class. Results indicate that our machine learning models show high predictive performance (Area Under the Receiver Operating Curve greater than 90). This suggests that the weather around a lookout tower is strongly associated with its technical efficiency classification. We also develop a method to combine the feature importance of various machine learning models to offer an overall rank of the predictive power of each weather variable. We refer to this method as the AUROC Weighted Permutation Feature Importance. While all the meteorological variables we consider play a role in predicting the technical efficiency of a lookout

tower, our model weighted measure indicates that humidity is the most important classification feature, followed by incoming radiation (second), wind speed (third), air temperature (fourth), precipitation (fifth), and finally snow water (sixth).

1.2. Scope of the Work

It is important to recognize that this work focuses on productive efficiency. The technical efficiency score compares detection output against an output benchmark determined by the production frontier. By design, the model allocates lookout towers to the detection frontier, or below it. As such, the model produces two important pieces of information: i) how many lookout towers are “frontier” towers and ii) for those lookout towers below the production frontier, on average, how far are they to the frontier. In both metrics, as discussed above, we find that the Alberta detection system performs extremely well with most towers at the technical efficiency frontier. Tower that are below the frontier are really close to it. On average, only a 5.9% increase in output detection is required to reach the frontier. This level of average inefficiency is much lower than typical DEA estimates reported in the literature. For example, in studying the technical efficiency of avoiding forest fire damage in European Union countries, Gutiérrez and Lozano (2013) find an average level of technical inefficiency ranging from 23.7% to 41.9%.

Additionally, this work develops machine learning models to predict whether a lookout tower can be classified as technically efficient or not based on local weather variables. We find that most of the ML models trained in this thesis exhibit high classification performance. This implies that the weather around a lookout tower is a good predictor of technical efficiency

classification. Since weather is a non-discretionary factor (also referred to as environmental variables, refer to Simar and Wilson (2007)), the high performance of ML models incorporating weather suggests that there is limited room for other potential discretionary drivers of technical efficiency. While this represents a challenge for policy action, perhaps this result should not be surprising given that the level of technical efficiency in Alberta's wildfire detection system is already very high.

This research shows how production economics can be applied to wildfire detection. It is important to acknowledge production reflects only one side of an economic system. As such, this work does not examine general equilibrium impacts. The analysis here does not serve as a comprehensive evaluation of the detection system and is not intended to be interpreted as a benefit-cost analysis or an analysis of the economic efficiency of detection resources. In fact, it is pretty obvious that Alberta's detection system produces great value for the Albertan public. For example, the annual budget allocated to Alberta's lookout towers is approximately 1.4 million dollars (MNP LLP 2020). This value pales in comparison to the potential of damages of a single escaped fire. For instance, estimates from the recent wildfires in Hawaii value damages at approximately \$6 billion US dollars (CNN Business 2023).

Finally, we note that this work is subject to data limitations. For instance, we have limited information about detection inputs and therefore the focus is on technical efficiency of the visibility profile of lookout towers. Additional information such as lookout observer characteristics (e.g. experience and fatigue), smoke column variables (e.g. color, trajectory, and position relative to wind), and commercial airline routes can also be incorporated in a production framework to measure technical efficiency of detection. Other limitations of this work are discussed in section 7.1.

1.3. Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 offers additional insights on the data. Chapter 3 describes key statistics of Alberta's wildfire detection program focusing on the role of lookout towers. Chapter 4 develops the technical efficiency framework, including model specification and the estimation approach. Chapter 5 presents the technical efficiency results. Chapter 6 discusses applications of machine learning in the field of wildlife detection and presents the ML models employed to: i) classify towers as technically efficient (or not) and, ii) measure the importance of various weather variables in such a classification. Finally, Chapter 7 offers concluding remarks, with a discussion of the results and limitations of the work.

Chapter 2. Data

This work leverages information from three datasets. The first is a fire-level dataset provided by AB Wildfire. The fire data contains information that explains fire characteristics from 2006 to 2021. This includes variables such as the fire code used to identify each fire, fire start geo-coordinates (latitude and longitude), and the agent detecting each fire. Additionally, the data contains the total area burned (when fire is extinguished). The dataset also provides time variables such as fire start and reporting dates and times. There are a total of 23,174 wildfires that were reported during the time-period chosen for the study, however, due to missing data and outliers (wildfire whose detection took longer than two weeks), the final working sample contains 21,269 wildfires.

Table 2.1 shows summary statistics of fire-level variables. The table shows that unplanned detection accounts for 42% of fires in our sample. Lookout towers show the highest frequency out of all planned detection agents and account for 31% of wildfires detection. Average reporting delay (the difference between fire start time and the time the fire was reported) is approximately 12.4 hours, but with significant variance (standard deviation of 38.6 is more than 3 times larger than the mean). Finally, the average area burned is approximately 150 ha. The distribution of this variable is extremely skewed with 14,061 small fires (class A, ha < 0.1) and only 319 large fires (Class E, ha > 200).

Table 2.1 Summary Statistics of Fire-level Variables

Variable	Mean	Std. Dev.	Min	Max
Detection Agent				
Air patrol	0.110	0.313	0	1
Ground patrol	0.161	0.368	0	1
Lookout tower	0.308	0.461	0	1
Unplanned	0.421	0.494	0	1
Reporting delay (hours)	12.377	38.621	0.0	335.317
Area burned (ha)	150.6	5682.3	0.01	577,646.8

Notes: Sample of 21,269 fires.

The second dataset is at the tower-level. This lookout tower data contains tower characteristics such as the unique tower identifier, tower geo-coordinates, and the range of towers. Furthermore, AB Wildfire offered visibility analysis maps for each lookout tower. These viewshed analysis grid plots are utilized to categorize whether a section of land (25-by-25 meter squares) falls into visibility types of “directly visible”, “indirectly visible”, “screened”, or “blind” (further details are provided in section 4.4). Table 2.2 shows summary statistics of the visibility profile of the 127 lookout towers that were active (produced at least one detection in a given year) in the period 2006 – 2021. On average, an observer is able to directly visualize 557 square kilometers of the lookout tower’s surveillance region.

Table 2.2 Summary Statistics of Tower-level Variables

Variable	Mean	Std. dev.	Min	Max
Area (km sq)				
Directly visible	557.3	322.8	55.9	1,645.2
Indirectly visible	1,831.9	830.4	71.8	3,698.8
Screened	1,146.5	496.0	76.9	2,934.6
Blind	1,452.4	1,039.1	8.4	4,527.6
Total Area	4,988.0	127.9	4,060.6	5,026.5

Notes: Sample of 127 lookout towers

The third dataset contains weather station observations collected and maintained by ACIS. ACIS is a government agency that manages climate and weather databases in Alberta. Weather data are collected using 501 meteorological stations placed around the province. A map of the locations of these weather stations is provided in Appendix G. The meteorological weather stations observations captured at an hourly level are interpolated across 6,900 townships in Alberta. Townships are 6-by-6 mile squares (93.240 square kilometers) that form a grid across the province. ACIS uses inverse distance weighed interpolation to obtain weather data for each township center. The nearest neighbor rule applies if no stations with non-missing observations are found within a certain radius from a township center (Alberta Climate Information Service 2019).

Table 2.3 offers summary statistics of weather variables reported at the township-month level, from 2006 to 2021. For example, average wind speed is approximately 10 kmh^{-1} , monthly accumulated precipitation is on average 9.7 mm, and maximum temperature is, on average, 8.6°C .

Table 2.3 Summary Statistics of Weather Variables

Variable	Obs.	Mean	Std. dev.	Min	Max
Wind speed (kmh^{-1})	896,027	10.30	3.73	0.29	48.23
Precipitation (mm)	935,802	9.74	15.04	0.00	562.72
Snow water (mm)	935,802	25.78	45.05	0.00	1448.50
Incoming radiation (MJ m^{-2})	1,141,582	394.95	236.26	27.38	939.57
Humidity (%)	1,080,647	70.62	9.00	37.00	98.71
Max air temperature ($^\circ\text{C}$)	1,141,582	8.64	11.67	-23.13	33.68

In the next chapter, we offer insights into the entire detection system, emphasizing the significance of lookout towers and the important role they play in wildfire detection.

Chapter 3. Lookout Towers and Wildfire Detection

This section offers insights on Alberta's wildfire detection program by describing key wildfire statistics from our data during the period of 2006 to 2021. Alberta is a large province with an area of 39 million hectares. The goal of AB Wildfire is to protect the designated Forest Protection Area (FPA) that accounts for two-thirds of the entire province. A reference map of Alberta's FPA is included in Appendix A (Alberta Wildfire and FireSmart 2020).

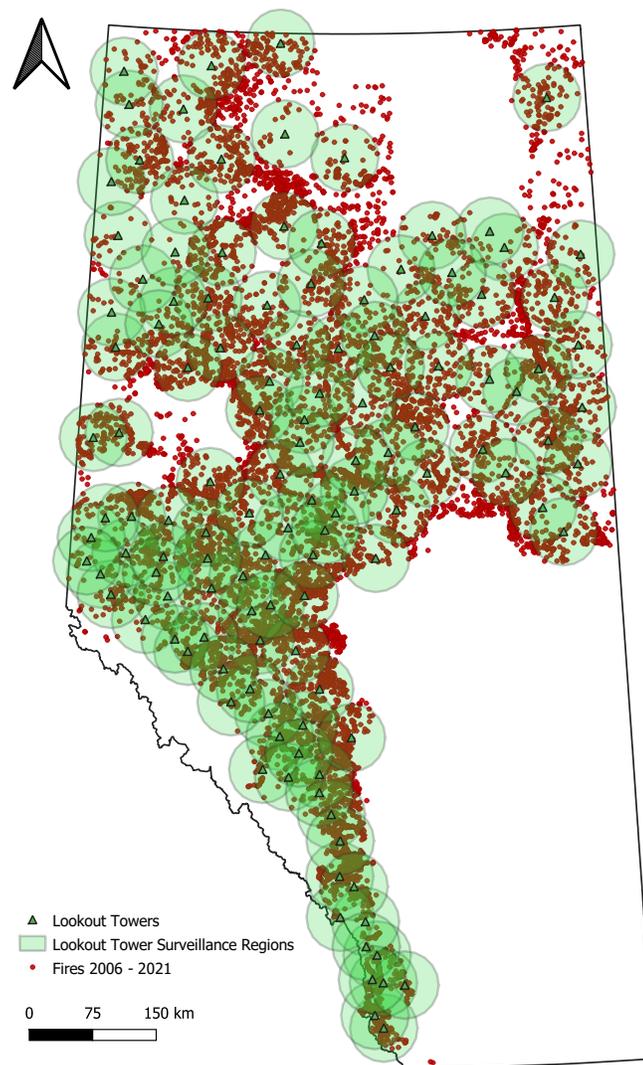


Figure 3.1 Fire Locations and Lookout Tower Surveillance Regions in Alberta

Figure 3.1 plots the spatial distribution of wildfires. The red dots represent 23,174 wildfires, the green triangles represent the 127 lookout towers in our data, and the green circles represent the surveillance region (SR) of each lookout tower. The figure includes a significant amount of white space, this area corresponds to regions outside the mandate of AB Wildfire. The focus is on protecting the designated FPAs, therefore, the detection program is contained within it. Wildfires in our data by definition have a point of origin inside the FPA (AB Wildfire 2020, p.1). No towers exist outside this area and no fires are reported in these regions. Any fires outside the FPA are monitored by the respective county or municipality.

Fires reported in Figure 3.1 have a size classification. Fire size class represents how big a wildfire is in terms of the final area burned which is measured in hectares. A fire class may change as the fire burns and gets bigger until the final burned area is estimated. Therefore, the final size is assigned after the given fire is completely extinguished. This variable can take the values of A, B, C, D, or E based on the following thresholds: fires of size less than or equal to 0.10 hectares (ha) are of size class A. Fires larger than 0.10 ha to 4.0 ha are of class B. Fires more than 4.0 ha but smaller than 40.0 ha are class C. Fires larger than 40.0 ha till 200.0 ha are class D, and any fire more than 200.0 ha in area is of size class E (AB Wildfire 2020, p.12).

Figure 3.2 presents the number of fires grouped by size class over time. The data shows a downward trend in the number of fires over time. Fires less than 4.0 hectares, i.e., class A and B, are most common. 67.211% of fires are classified as size A and 25.411% are classified as size B. The remaining 7.378% are fires larger than 4.0 hectares and are either of size C, D, or E. The small proportion of large fires indicates that most fires are detected early when they are small and pose lower risks. This aligns with the goal of effective detection described by AB Wildfire as, “report all smokes before reaching a size of 0.10 ha or less” (Lookout Observer Manual 2022, p.21).

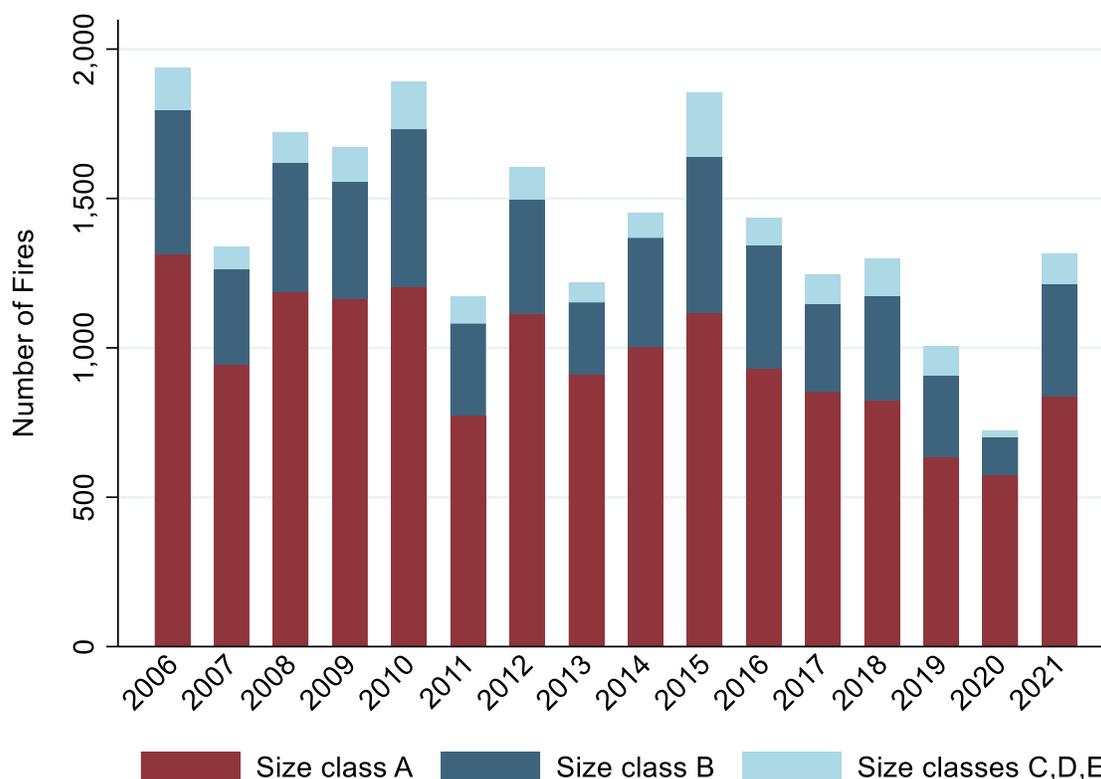


Figure 3.2 Frequency of Fires by Size Class

Table 3.1 offers insights on detection operations by reporting the number and proportion of fires detected by the four agents. The table includes the relative contribution as a percentage of all fires detected by each detection agent across three intervals during the 16-year period: 2006 to 2010, 2011 to 2015, and 2016 to 2021.

Table 3.1 Detection Frequencies of Different Detection Agents over Time

Detection agent	2006 - 2010		2011 - 2015		2016 - 2021	
	Freq.	Percent	Freq.	Percent	Freq.	Percent
Unplanned (UNP)	3,572	41.724	2,917	39.942	3,236	46.051
Lookout Towers (LKT)	2,766	32.309	1,994	27.304	1,960	27.892
Ground Patrol (GRP)	1,247	14.566	1,622	22.210	1,099	15.640
Aerial Patrol (AIR)	976	11.401	770	10.544	732	10.417
Total	8,561	100	7,303	100	7,027	100

Over the three intervals, unplanned detection agents (UNP) have the highest frequency by detecting most wildfires. There is a recent increase from the second interval to third interval. This agent costs AB Wildfire only 0.2% of the total spendings during the fiscal period 2015 – 2019 (fire year 2016 – 2020). The high participation by unplanned detection agents in Alberta is consistent with other wildfire agencies in different regions. Public reporting of wildfires is the most common detection agent in different countries. For example, public reporting contributed to 76.20% of all detections in Portugal and 55.90% in Spain (Rego et al. 2013). McFayden et al. (2019) find that 48.90% of all reported fires in Ontario are via public reporting during the period 2011 to 2018. This is consistent with Alberta (40.25%) in the same period. Thus, signaling that the high contribution of public reporting of wildfires is a trait not exclusive to our dataset.

Lookout towers detect 29.357% of fires across all periods. Detections by lookout towers experience a 5.005 decrease in percentage points from the first to the second interval and an increase of 0.588 percentage points in the following interval. Lookout tower expenditures account for 43.070% of the total expenditure in the fiscal year 2015 to 2019. The decreasing trend of lookout tower detections from the first interval (2006 – 2010) to the second interval (2011 – 2015) raises concerns.

Ground patrols detect 17.334% of fires on average. The trend in their contribution increases from the first period to the second (7.644 percentage points) then decreases in the third period (6.57 percentage points). Note that ground patrols play a significant role in detecting wildfires, particularly those related to residential and recreational areas. This is partly because there are many wildfires in populated regions that are quickly identified and reported (MNP LLP 2020). As a result, GRP shows a higher contribution, however, this may not necessarily correspond to a higher impact to the wildfire management system (MNP LLP 2020).

Aerial patrol is the least active detection agent employed by AB Wildfire while being the most expensive detection method. Note that apart from specific aerial flights for patrolling, air patrols act as a backup to the lookout tower system by focusing on areas that are screened or blind (Lookout Observer Manual 2022, p.22). During the fiscal years 2015 to 2019 AB Wildfire spent a total of 16.96 million dollars on their detection program, the cost of aerial patrol represents 56.830% of the total spending (MNP LLP 2020).

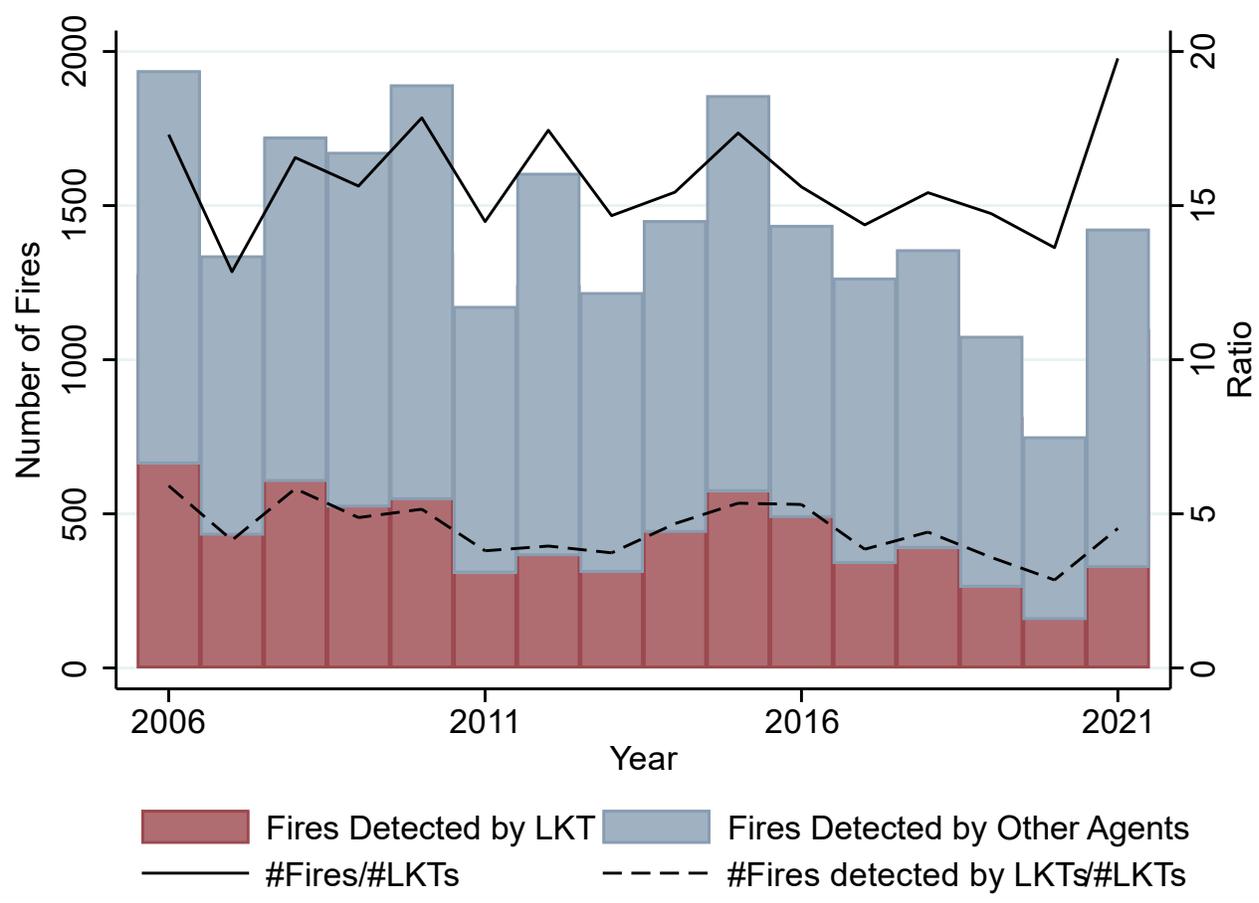


Figure 3.3 Number of Fires and Contribution of Lookout Towers over Time

Figure 3.3 illustrates the contribution of lookout towers for detecting wildfires in Alberta. It includes a stacked bar graph with the red bars representing the number of fires detected by lookout towers and the blue bars representing fires detected by all the other agents (UNP, GRP,

and AIR). The y-axis on the left-hand side represents the yearly frequency of fires. For a better insight on the performance of lookout towers, the figure is overlaid with two lines. These lines represent different ratios and are propagated within the dataset. The y-axis on the right-hand side labelled 'ratio' is the axis of reference when focusing on the two trends lines.

The solid line represents the ratio of the number of fires to the number of active lookout towers, by year. A higher ratio shows that lookout towers carry, on average, a larger share of fire detection. From an economics perspective, this can be interpreted as a proxy for the demand for lookout tower detection. The data shows that the ratio of fires per tower exhibits a slight downward sloping trend from 2006 to 2021.

The dotted line is as proxy for the average productivity of the existing lookout system. It represents the ratio of fires detected by lookout towers by the number of active lookout towers in a year. This ratio is associated with the productivity of lookout towers in the detection system. The data shows that this trend is decreasing over time.

Additionally, our data also allows us to examine wildfires that originated outside the range of lookout towers. We find that 3,485 wildfires in the 16-year period (approximately 218 fires per year) are outside the surveillance region of all lookout towers. These fires can be seen in Figure 3.1 as the red dots that lie outside the surveillance regions (green circles). Although observers focus on their surveillance region, fires outside these regions can also be detected by lookouts. However, lookout towers detect fires inside their surveillance region faster than the ones that are past their range. Thus, fires that are not in the surveillance region of towers may have a higher chance of going undetected from the time they start, especially in remote areas with very little

human presence. Exploring the data and looking at fires that do not lie inside a surveillance region can give us insights on how the detection program handles these fires.

Figure 3.4 plots the frequency of fires outside the surveillance region of towers and shows how many of these fires were detected by the different detection agents. The total number of fires outside the surveillance region of towers is increasing over time, which is concerning. Figure B.1 in Appendix B provides insight on these wildfires based on their size classes. Data in Figure 3.4 show that there is a decreasing trend in the number of fires detected by lookout towers, but it is important to consider that the number of active lookout towers are reducing over time. Similarly, aerial patrols are detecting fewer fires outside the surveillance region during the 16-year period. On the other hand, ground patrols and unplanned agents show an upwards trend over time.

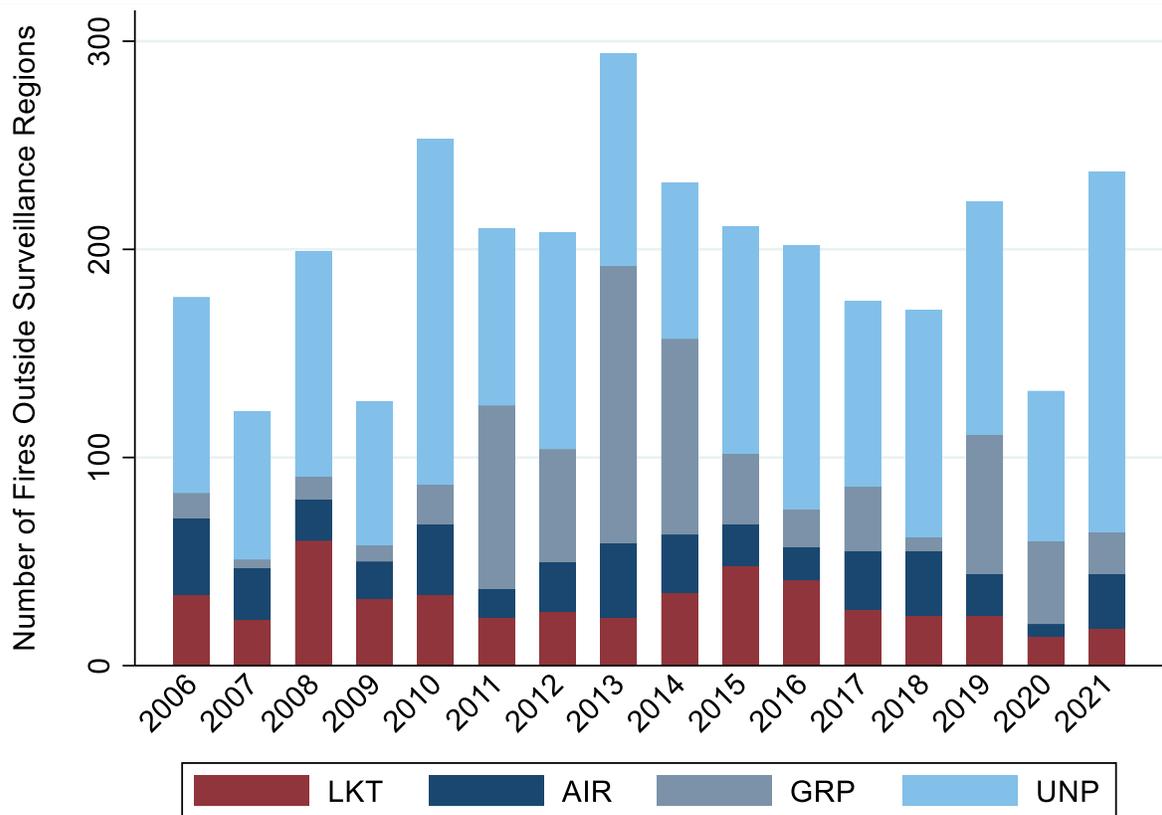


Figure 3.4 Frequency of Detections outside Surveillance Region

Chapter 4. Technical Efficiency Framework

4.1. Technical Efficiency

The theoretical framework underlying the analysis conceptualizes a wildfire detection system where features of lookout towers are inputs (x) in the production of fast and reliable wildfire detection, i.e., detection outputs (y). In classical microeconomics, production is fully described by a production function, $y = f(x)$. In reality, lookout towers with similar (or the same) input levels may produce varying levels of wildfire detection. Our framework accounts for this possibility by utilizing a set theoretic representation of wildfire detection technology. Specifically, we consider an output production set (also known as technology set) defined as:

$$P(x) = \{y: x \text{ can produce } y\}, \quad (1)$$

i.e., the set of all outputs (y) that can be produced using inputs (x). In other words, the set of outputs that are technically feasible given the inputs.

Figure 4.1 depicts a production set with two outputs in the output space. The boundary of the production set is the production (or technology) frontier and defines the detection system's production possibility set, where P is a closed set. The feasible possibilities of output combinations are represented as the blue shaded region. The grey area represents any point outside the production possibility set which is unfeasible. The black dots represent examples of different production units (e.g., lookout towers) and their output combinations. It is typical to assume that P is convex, with

free disposability of inputs and outputs and weak essentiality – units can choose to produce nothing, and non-zero level of outputs cannot exist with zero levels of inputs.

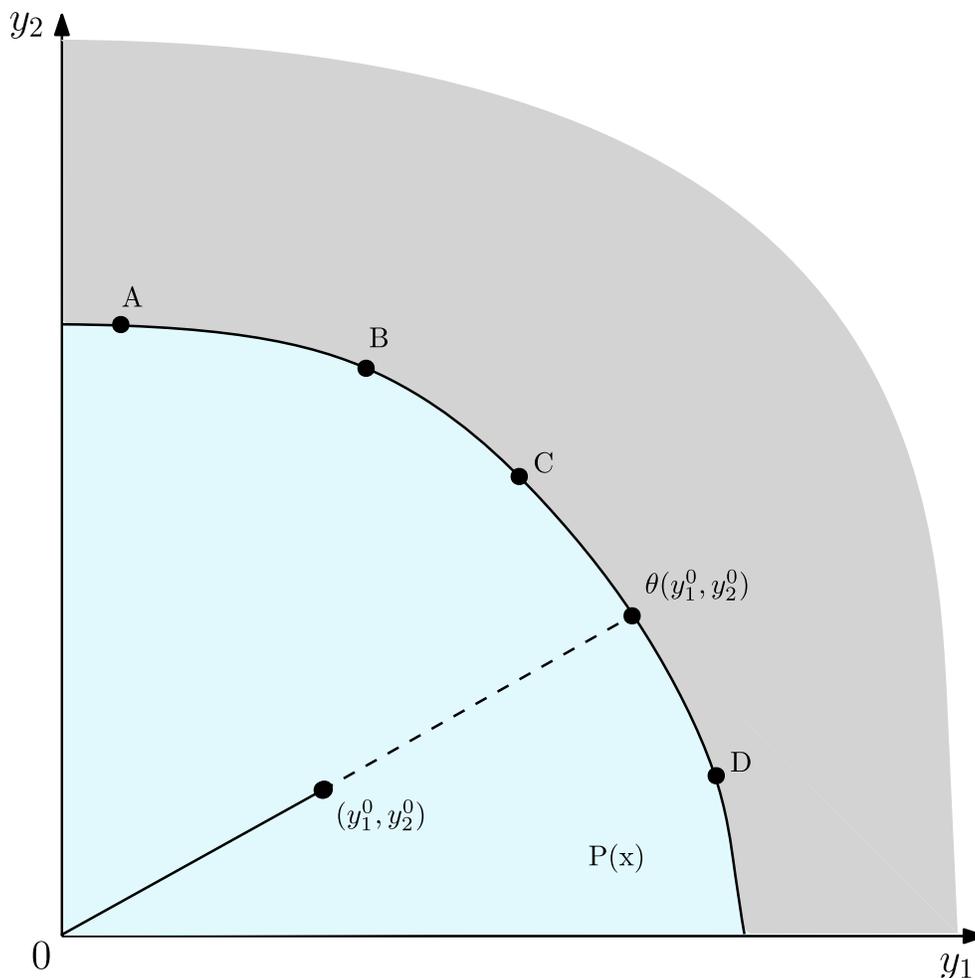


Figure 4.1 Production Possibility Set and Distance Functions

Convexity of the production set is a fundamental concept in production economics (Coelli et al. 2005; Chambers 1988). The assumption implies ‘no free lunch’: improvements in one output cannot be possible without adjusting the input vector or levels of another output. In input-space, this mathematical property ensures that no production unit can increase outputs without increasing their input levels. Moreover, a convex production set results in a concave production frontier in the output-space, which aligns with the notion of decreasing returns to scale in wildfire detection.

For a given technology level, the production frontier represents the maximum output that can be achieved from each input level. Hence, this frontier provides a natural feasible benchmark for production units. Malmquist (1953), Shephard (1981), and Farrell (1957) develop the concept of distance functions, which formalizes by quantifying a benchmark measure for productive performance as the scalar θ such that:

$$d(x, y) = \max\{\theta: \theta y \in P(x)\}, \quad (2)$$

The output of the distance function (θ) is known as the output-oriented technical efficiency of the production plan (x, y) , with $\theta \geq 1$. This measure answers the following question: “How much can output quantities be proportionally expanded without changing the input quantities used?” (Coelli et al. 2005, p.54). In other words, given a fixed input level, what is the maximal and feasible proportional expansion, or what proportional expansion makes a production unit technically efficient. Figure 4.1 illustrates this concept. The point with output combination (y_1^0, y_2^0) corresponds to a technically inefficient lookout tower. θ represents the maximum factor by which this lookout tower can radially increase detection output and therefore be pushed to a technically efficient point $\theta(y_1^0, y_2^0)$, holding the inputs fixed. Therefore, production units that are technically efficient have an efficiency score of 1 and technically inefficient units are assigned a score of more than one.

This measure can be interpreted in another way using fractions. What scalar (technical efficiency score θ) when multiplied by the technically inefficient production plan (y_1^0, y_2^0) , will proportionally expand the lookout tower to the boundary of the production possibility set, deeming it technically efficient:

$$\theta(y_1^0, y_2^0) = (y_1^{\text{efficient}}, y_2^{\text{efficient}}),$$

Where $(y_1^{\text{efficient}}, y_2^{\text{efficient}})$ is a fully efficient (technically efficient and on the production frontier) production plan. The equation above can also be represented as follows:

$$\theta = \frac{y_1^{\text{efficient}}}{y_1^0} = \frac{y_2^{\text{efficient}}}{y_2^0}.$$

4.2. Parametric vs Nonparametric Estimates of Technical Efficiency

Empirical measurement of technical efficiency score, θ , requires an estimation of the production frontier. There are two main methodologies: parametric and non-parametric, or Stochastic Frontier Analysis (SFA) and Data Envelopment Analysis (DEA), respectively. The two differ in the incorporation of random errors, distribution assumptions of the random errors and inefficiencies, and restrictions on functional forms of the production frontier (Cooper W. William, Seiford and Zhu 2004).

SFA was introduced by Aigner et al. (1977) and Meeusen et al. (1977). It explicitly defines a production function and uses econometric techniques to estimate its parameters. This methodology calculates technical inefficiency by examining deviations of production units from the parametric production frontier. The deviations are either due to technical inefficiency, stochastic random errors, or statistical noise in the data. Incorporating errors or disturbance terms was introduced by Schmidt and Lemke (1976). This method also defines the theoretical statistics needed for inference and hypothesis testing (Aigner et al. 1977). However, parametric approaches have an important weakness. They force a functional relationship between inputs and outputs,

which creates a bias if the functional form is misspecified (Dhungana, Nuthall and Nartea 2004; Schmidt 1985). This can be a serious issue because the true production technology is unknown and often highly nonlinear.

Non-parametric methods require fewer assumptions regarding the production technology (Cooper W. William et al. 2004). DEA, first introduced by Charnes et al. in 1978, provides a framework to measure the technical efficiency of production units (Hjalmarsson, Kumbhakar and Heshmati 1996). It is a fully nonparametric approach and does not assume any functional relationships between inputs and outputs, thus, minimizing the risk of an incorrectly specified functional form (Schmidt 1985). It only requires data on the production activities which can be easily accessed and applied. No data manipulation is needed such as assigning weights to variables, and it is easy to incorporate production technologies that use multiple inputs and outputs (Homburg 2001). Conceptually, DEA finds the production set by constructing the smallest convex set that ‘envelops’ the data. Empirically, it does so by solving a linear programming model that constructs a piece-wise frontier based on input and output observations (Coelli et al. 2005; Simar and Wilson 2000). Appendix C discusses the DEA programming problem we use in our analysis. Once the frontier is estimated, DEA calculates technical efficiency scores using distance functions (Aldamak and Zolfaghari 2017).

4.3. Bias-corrected DEA Scores

Simar and Wilson (1998) develop a statistical treatment for DEA-based technical efficiency estimates. Despite the fact that DEA estimates of technical efficiency are computed from

mathematical programming techniques, Simar and Wilson demonstrate that, conceptually, DEA scores are measured relative to an *estimate* of the unknown true production frontier. They argue that nonparametric estimators such as DEA are based on a finite sample of inputs and outputs from production units. As such, they propose a data generating process that allows for the examination of sampling variations of DEA technical efficiency scores.

While Korostelev, Simar and Tsybakov (1995a, 1995b) show that, under weak general conditions, the DEA is a consistent estimator of the true production set, Simar and Wilson (1998) emphasize that DEA scores are biased in finite samples and develop a method to correct for the finite-sample biases.

To understand the source of the bias, note that the envelopment frontier is heavily dependent on the sample. The observed sample can be thought of as being drawn from the true data generating process. While an infinitely large number of draws would result in a full depiction of the production frontier, finite samples do not capture the true production frontier in its entirety.

Figure 4.2 illustrates this bias in the output-space. Consider an industry with the true (or theoretical) frontier given by the solid black line. The points A, B, C, D, E, F, G, H, and I represent production plans. Production plans A-F lie on the production frontier, while points G, H and I lie below the frontier. Now, consider a situation where the researcher only observes a sample represented by the red dots i.e., A, B, E, F, G, H, and I. Note that this sample does not include two data points (C and D). As we will discuss next, these data points are important because they represent radial benchmarks of the observed production units. Point C is should be the radial benchmark for unit H, and point D is the proper benchmark for unit I.

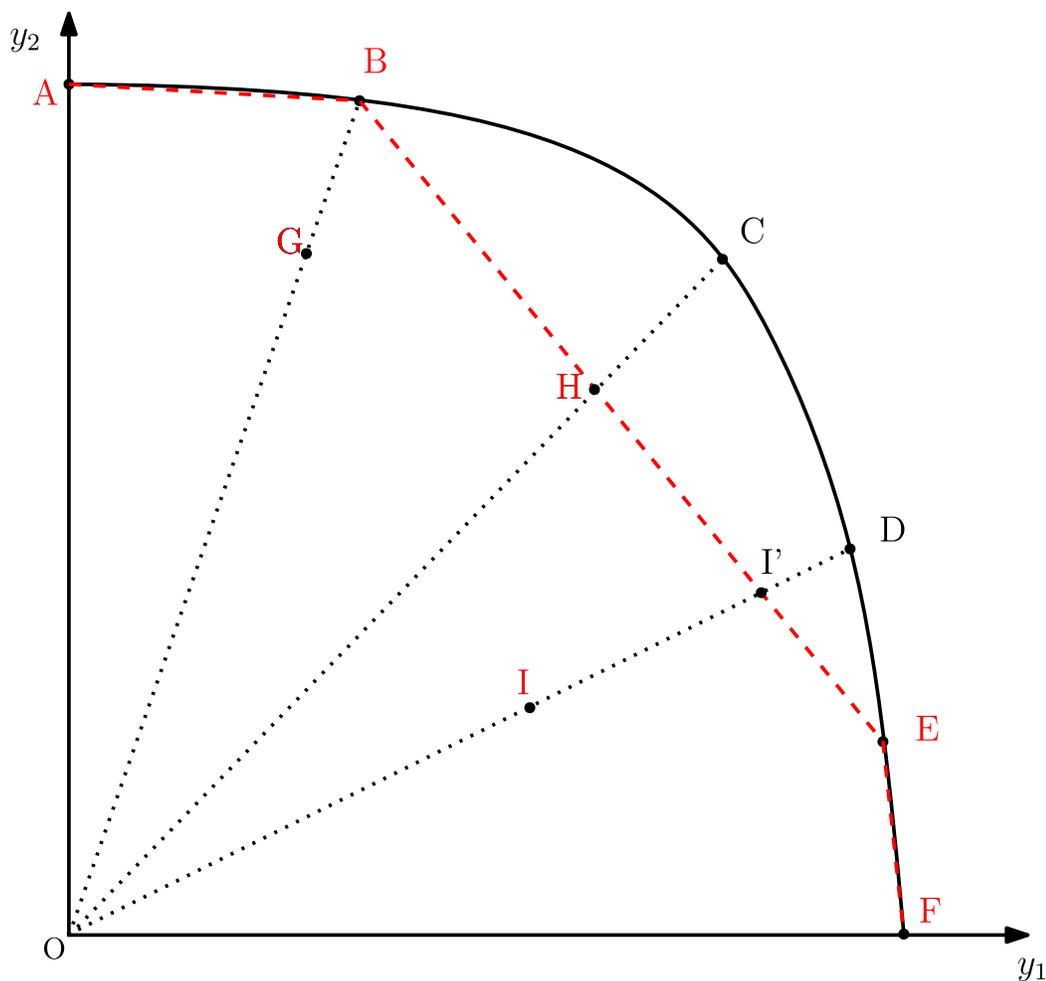


Figure 4.2 Bias in Constructing DEA Frontier

The red dotted line is one example of the frontier constructed when the researcher applies the envelopment methodology to the observed production plans. This frontier is the DEA frontier applied to this specific sample. The DEA estimates for production units with production plans A, B, E, F, and G will be estimated correctly, without a bias. The sample points A, B, E, and F lie on the true frontier and therefore, for these points, the envelopment frontier matches the true production process. As a result, their DEA scores (θ) are correctly estimated with a value of 1. Even though the observed point G does not lie on the true frontier, point B (which is the true radial benchmark for G) is in the sampling data. Thus, the estimate of the technical efficiency score for

G will not contain a bias. In this case, DEA will assign a technical efficiency score less than 1, thus correctly and without biases deeming G technically inefficient.

However, a bias exists in the DEA scores for points H and I. Point H, which is a technically inefficient production plan, is observed in the sampling data while its true radial benchmark C is not observed. Therefore, the DEA frontier is constructed using point H. Unit H is considered to be technically efficient by the envelopment method when, in reality, it is technically inefficient. Thus, the DEA score of technical efficiency for point H will be biased upwards (DEA overestimates technical efficiency). A similar type of bias affects the DEA's estimation of technical efficiency for unit I. Point I' is not a sampling point, it is a convex combination of points H and E, constructed by DEA to serve as the radial benchmark for sample point I. This radial benchmark is not the true radial benchmark as I' does not lie on the true frontier. The true radial benchmark for point I is point D, but this is not present in the researcher's sample. Using the distance function, we can see that the true value of technical efficiency for point I is OI/OD but DEA assigns again a larger value of OI/OI' .

As mentioned above, DEA scores of technical efficiency is not computed using the true frontier and sample variation affects the estimated frontier (Long et al. 2020). Works in this literature have shown how an understanding of sampling variations can be used to address the bias of DEA technical efficiency measurements. Simar and Wilson (1998) develop a bootstrapping method that accounts for and corrects the DEA bias in technical efficiency estimates. It can be applied to multi-input or output models and provides a statistical treatment to envelopment estimators allowing for the construction of confidence intervals and for the correction of the DEA bias (Simar and Wilson 2000).

In short, bootstrapping is a statistical method that generates pseudo sub-samples that are drawn with replacement from the sampling dataset with the intention of testing the reliability of estimates³. It provides an approach to examine statistical properties of estimations and it can be a valuable tool when analytical solutions are not available (Abatania, Hailu and Mugeru 2012). Bootstrap resampling is typically conducted a few hundred times (for the purpose of bias-correcting technical efficiency scores, we use 399 repetitions). The estimations from all iterations together form a distribution that can be used for statistical inference and hypothesis testing (Long et al. 2020).

We use figure 4.3 to explain the general intuition of Simar and Wilson's bootstrapping method of bias correction. The solid line is the true production frontier. The frontier represented by the black dashed line going through production plans A, B, C, D, E, and M is the DEA frontier. As discussed above, DEA overestimates technical efficiency as the envelopment of the sample underestimates the true production set. The grey shaded region represents the deviation between the true frontier and the DEA frontier. Applying the bootstrap approach resamples the observed data. The figure illustrates one bootstrap iteration. In the figure, we assume the bootstrap iteration picks a sub-sample of points A, C, E, and M to construct a DEA frontier. This bootstrap DEA frontier is represented by the blue dotted line. The red region is the deviation between the DEA frontier and a bootstrapped DEA frontier.

³ Sub-samples can be picked in more than one way, please refer to Appendix D for a short description of different types of bootstrapping approaches.

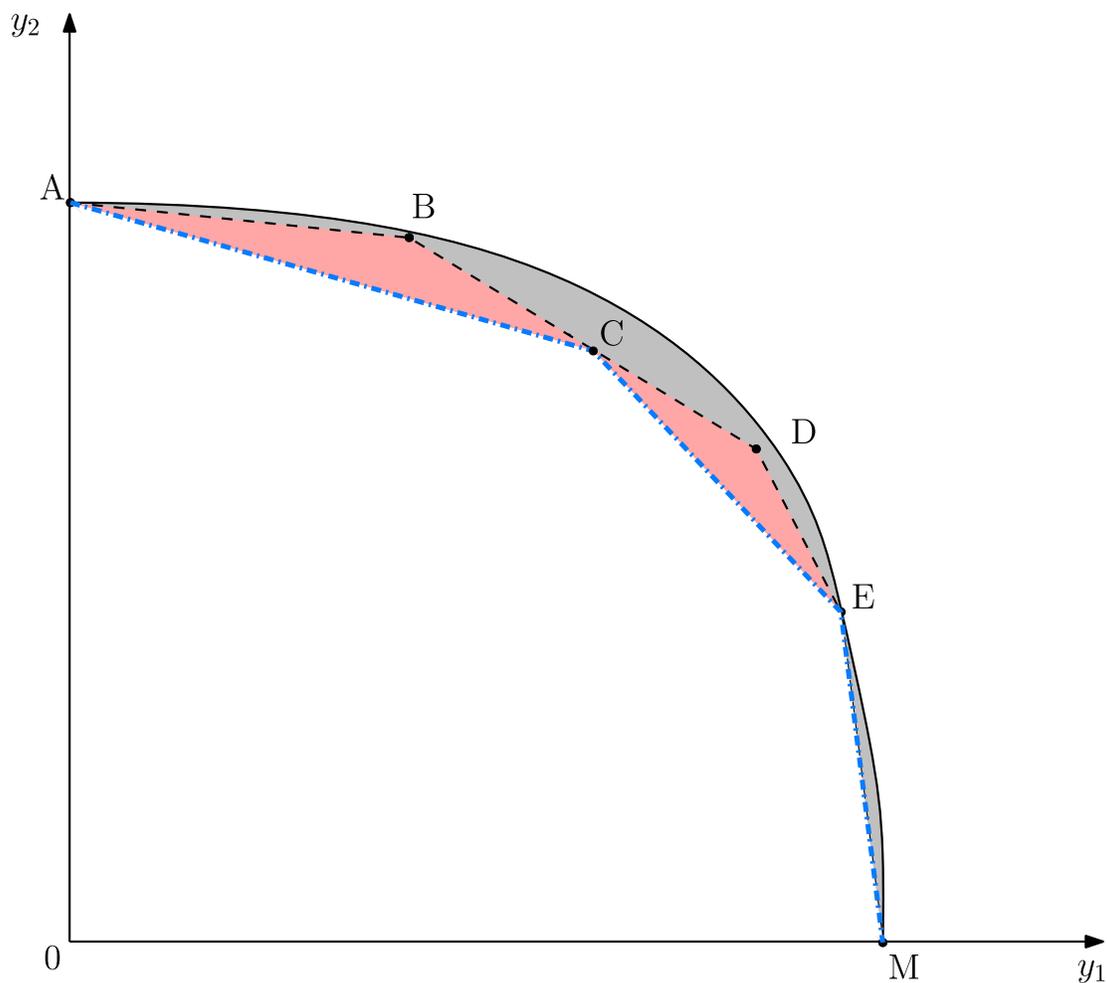


Figure 4.3 Theoretical Frontier vs DEA Frontier vs Bootstrap Frontier

Technical efficiency estimates are computed for each bootstrap sub-sample and the process is repeated several times. Each iteration picks a slightly different sub-sample; therefore, a different bootstrap frontier is used to calculate technical efficiency estimates at each iteration. The insight from Simar and Wilson is that the deviations between the bootstrap and DEA frontier can help account for the DEA bias in technical efficiency scores. The intuition behind this approach is that deviations between the bootstrap frontier and the DEA frontier mimic the deviations between the DEA frontier and the true theoretical frontier. This provides a bootstrap definition of the bias that can be used to bias-correct DEA estimates from the full sample.

Below we offer a summary of the technical efficiency framework developed by Simar and Wilson (1998). As the true production technology, $P(x)$, given by equation (1) is unknown, the DEA technical efficiency scores are given by replacing $P(x)$ with the envelopment estimate $\widehat{P}(x)$ in the distance function (2):

$$\hat{\theta} = \hat{d}(x_i, y_i) = \max\{\theta: \theta y_i \in \widehat{P}(x)\}.$$

The score $\hat{\theta}$ is an estimate of true and unknown technical efficiency score θ . As $\hat{\theta}$ depends on $\widehat{P}(x)$, the sampling properties of $\hat{\theta}$ (an estimator for true efficiency) depend on properties of the method used to generate $\widehat{P}(x)$. Simar and Wilson (1998, 2000) use the bootstrap approach to address the complex and possibly unknown sampling properties of envelopment estimators. Applying the concepts above to the bootstrap samples, the bootstrap technical efficiency score for a given iteration can be expressed as:

$$\hat{\theta}^* = \hat{d}^*(x_i, y_i) = \max\{\theta: \theta y_i \in \widehat{P}^*(x)\}.$$

where $\widehat{P}^*(x)$ is the envelopment of the bootstrap subsample. The advantage of the bootstrap is that sampling distributions of $\widehat{P}^*(x)$ are known since the DEA estimate $\widehat{P}(x)$ is known. Assuming that $\widehat{P}(x)$ is a reasonable estimator of $P(x)$, the bootstrap distribution should mimic the sample distribution of the estimators which in turn mimics the true unknown distribution. As a result:

$$(\hat{\theta}^* - \hat{\theta}) \sim (\hat{\theta} - \theta). \quad (3)$$

Note that the bias of the DEA estimator $\hat{\theta}$ is given by:

$$\text{DEA Bias} = E(\hat{\theta}) - \theta,$$

and the bias of the bootstrap estimate is:

$$\text{Bootstrap Bias} = E(\hat{\theta}^*) - \hat{\theta}, \quad (4)$$

It follows that, according to (3):

$$\text{DEA Bias} \sim \text{Bootstrap Bias.}$$

$E(\hat{\theta}^*)$ in (4) is estimated by the average of $\hat{\theta}_n^*$, which is the DEA score for bootstrap sample n , with $n = 1, \dots, N$ iterations.

$$E(\hat{\theta}^*) = \frac{\sum_{n=1}^N \hat{\theta}_n^*}{N} = \bar{\theta}^*$$

Therefore, the estimate of the bootstrap bias in equation (4) is:

$$\widehat{\text{Bias}} = \bar{\theta}^* - \hat{\theta}.$$

Finally, the estimate of the bias can be used to correct the technical efficiency estimates. The bias corrected DEA score is obtained by subtracting the bias from standard DEA estimates:

$$\begin{aligned} \text{Bias Corrected DEA Estimate} &= \hat{\theta} - \widehat{\text{Bias}} \\ &= \hat{\theta} - (\bar{\theta}^* - \hat{\theta}) \\ &= 2\hat{\theta} - \bar{\theta}^*. \end{aligned} \quad (5)$$

4.4. Wildfire Detection Technical Efficiency Model

Technical efficiency analysis requires the specification of outputs and inputs used in production. For our purpose of estimating technical efficiency scores, we calculate inputs and outputs using data from AB Wildfire. We specify a model where, in each fire season (year), lookout towers are productive units that produce two detection outputs using their visibility profiles as inputs.

In our approach of characterizing lookout tower outputs to measure detection productivity, we incorporate the extensive and intensive margins for lookout tower detections. The extensive margin relates to the notion of knowing if the planned detection system (lookout towers) is carrying out detections or not. Once we incorporate this, we can evaluate the performance of lookout towers further by measuring how long it takes lookout towers to detect wildfires, explaining the intensive margin of detection output produced by lookout towers. Therefore, our analysis of technical efficiency of lookout towers accounts for both the extensive and intensive margins of outputs.

The first output we consider is the proportion of wildfires detected (y_1). This variable is the ratio found by calculating the fires detected by a lookout tower divided by the total number of fires in its surveillance region. The surveillance region is a circular area that the tower is responsible for monitoring. This area is calculated for each lookout tower using viewshed visibility plots. We note that it is possible for a tower to detect fires outside its surveillance region. Therefore, we use the convention that towers with the number of fires greater than the number of fires in a tower's surveillance region have output, y_1 , normalized to one. Hence, y_1 measures a desirable outcome that can range from 0 to 1. Higher values are correlated with higher tower productivity and lower values penalize the output of lookout towers as it fails to meet its mandate of detecting wildfires.

If a lookout tower fails to report a fire in its surveillance region it increases the risk the wildfire will grow to a destructive level (Mendes, 2010). Improving the proportion of fires detected by a lookout tower (y_1) can reduce the number large wildfires (Rego et al. 2013). Figure 4.4 shows a scatter plot of the proportion of fires detected across all lookout towers, by year. The red line is the fitted linear trend and is sloping upwards signaling that lookout towers are detecting a higher proportion of fires in their respective surveillance regions over time.

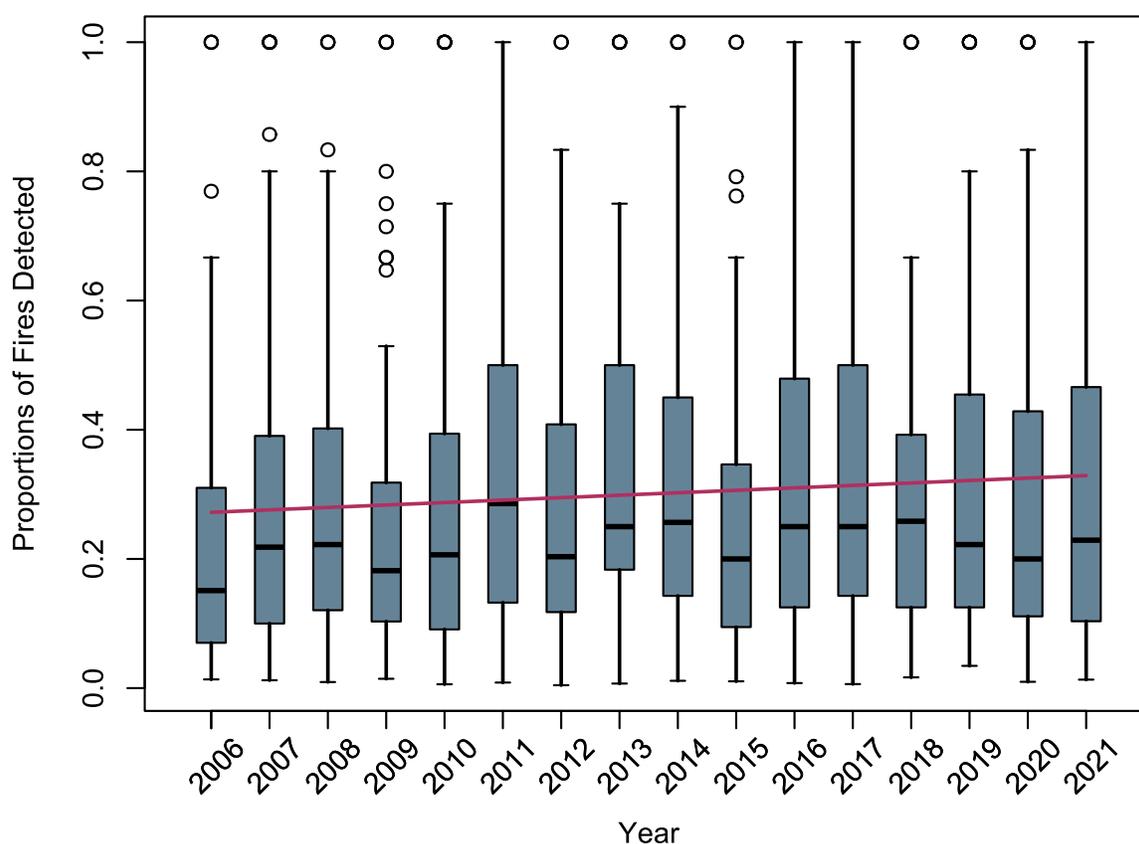


Figure 4.4 Proportion of Fires Detected by Lookout Towers

The second detection output captures the speed of detection, namely average early detection (y_2). To measure early detection, we first measure reporting delay as the time it takes for a lookout tower to detect fires. Measured in hours, reporting delay is the difference between fire start time and reported time. While technical efficiency analysis focuses on desirable outputs,

report delay is not a desirable outcome. Thus, we convert reporting delays into early detection by measuring the amount of time from detection to a later time threshold. We drop fires that were reported more than 2 weeks (336 hours) from the estimated fire start time as they may represent measurement errors and use 336 hours as the detection threshold. Therefore, average early detection measures the amount of time *before* 336 hours it took to detect a wildfire such that increases in y_2 represent increases in wildfire detection. For example, if a fire is detected 100 hours after its ignition, then $y_2 = 236$. If detection is faster, say 50 hours after ignition, then increases to $y_2 = 286$. An alternative specification relies on inverting undesirable outputs. However, the inverse function adds an artificial (nonlinear) variation in the data hence fundamentally altering the data distribution. These fire outputs must be aggregated to the level of lookout tower in a fire season. Therefore, y_2 is fire-level average early detection.

Figure 4.5 shows a scatter plot of average early detection with a linear trend line. On average, the speed at which lookout towers are detecting wildfires is increasing over time. Detecting wildfires in time increases chances of controlling them before they grow resulting in minimizing expenditure and firefighting resources. In fact, the timeframe from when a fire starts to when it is detected plays an important role in successful suppression (Castro, Akhloufi and Couturier 2018; Amiri et al. 2022) Additionally, early detection leads to smaller burned areas, reduced carbon emissions, and fire suppression costs (Scholten et al. 2021; Steele and Stier 1998).

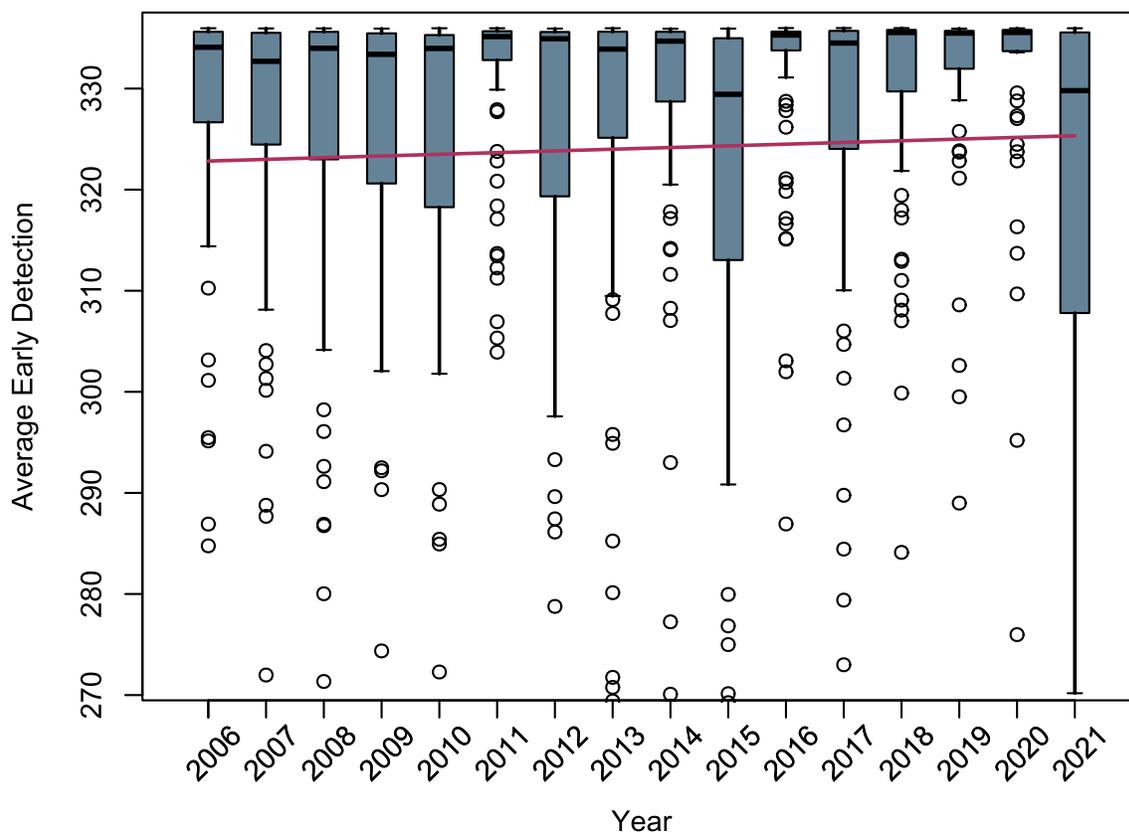
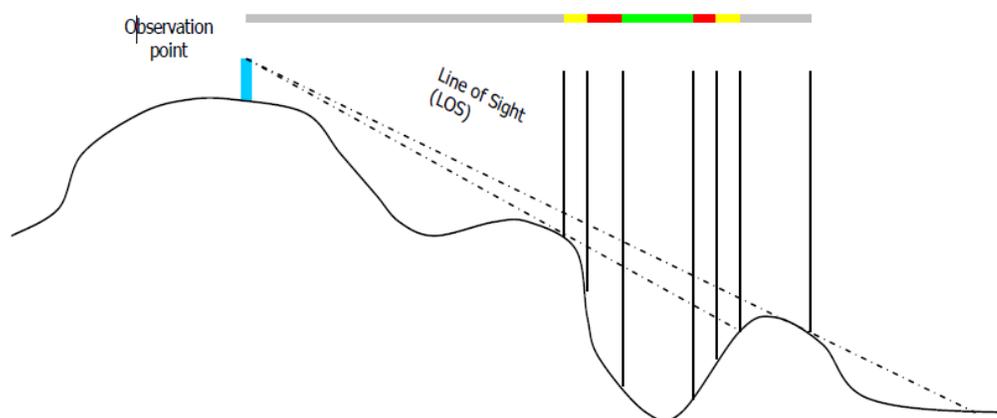


Figure 4.5 Average Early Detection Times Over Time

Lookout towers can vary significantly in their detection performance (Kucuk et al. 2017). Many factors affect how well a tower performs, such as, geographical characteristics around the lookout tower (Sakellariou, Sfoungaris and Christopoulou 2022; Çoban and Bereket 2020). Lookout towers are designed and placed in regions that allow for wildfire surveillance by a human observer. Therefore, visibility around a lookout tower is essential. Lookout tower observers use a colored map with a scale of 1:100,000 which are constructed using viewshed analysis. The analysis maps help view their area of responsibility which we refer to as the surveillance region (AB Wildfire 2022). The observer visualizes smoke and uses the station's instruments to map and report the fire position.



Lookout visibility classes

- **Visible** – 0 or more meters above line of sight.
- **Indirectly Visible** - up to 30 meters below line of sight.
- **Screened** – 30 to 100 meters below line of sight.
- **Blind** – greater than 100 meters below line of sight.

Figure 4.6 Line of Sight Viewshed Analysis for AB Wildfire Lookout Towers
(Source: Lookout Observer Manual 2022, p.164)

Figure 4.6 illustrates how the different visibilities are determined based on the line of sight from the observation point on a lookout tower to the landscape around it. Smoke seen from the point of view of the lookout tower observer can rise from a valley or beyond a hill or ridge causing the observer to not be able to see the ground (Lookout Observer Manual 2022, p.164). Therefore, it is crucial to know a visibility profile for a given tower. As such, following the line of sight (LOS) approach, areas around a tower are categorized into four distinct types. If the area is zero or more meters above the LOS, it is classified as visible (light grey). If it is up to thirty meters below the LOS, it is said to be indirectly visible (yellow). An area that is between thirty meters and a hundred meters below the line of sight is said to be screened (red) and any area that is more than a hundred meters below the line of sight is completely blind (green). In such a way, viewshed analysis of the surveillance regions is used to construct lookout tower visibility maps, as seen in Figure 4.7. The

visibility map of a lookout tower allows us to further assess its visibility profile. Using GIS tools, we compute the areas (in square kilometers) of the total surveillance region based on the four different visibility areas.

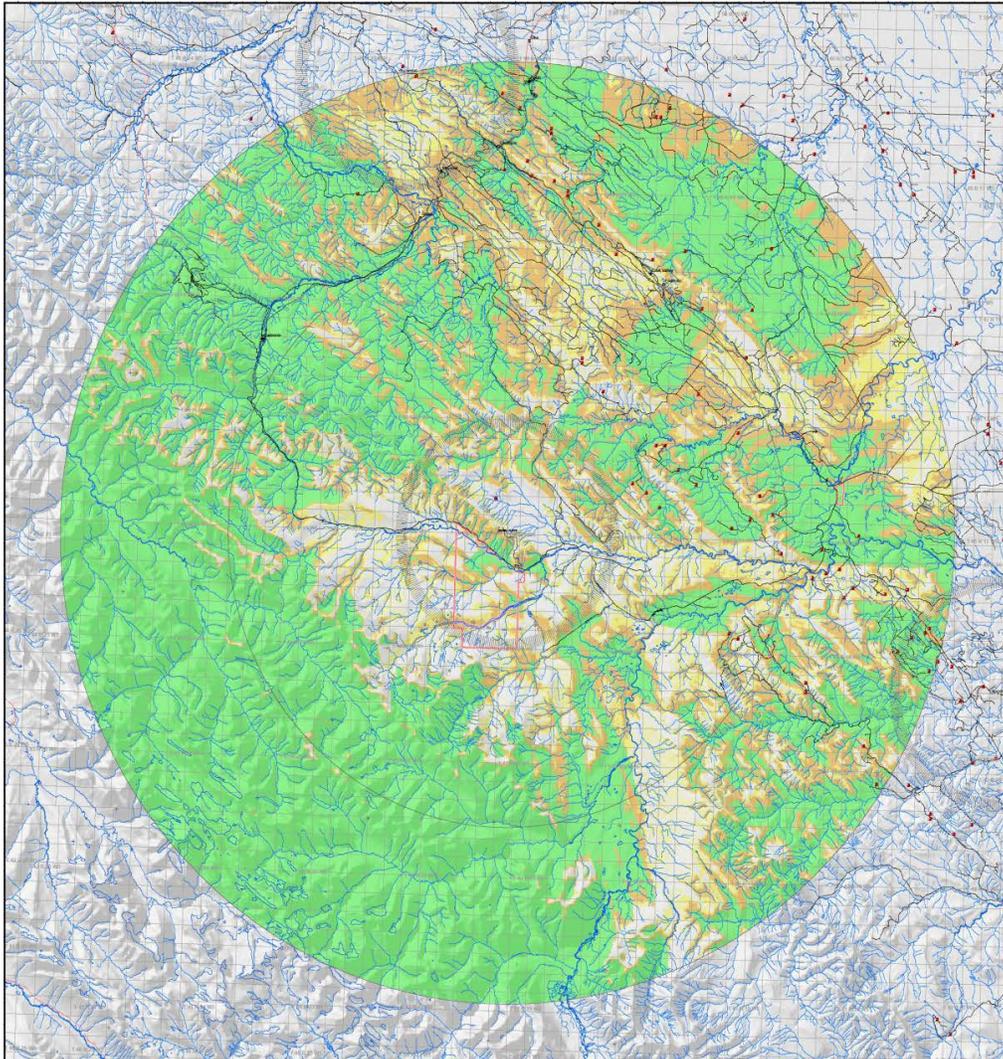


Figure 4.7 Lookout Tower Visibility Map
(Source: Lookout Observer Manual 2022, p.8)

It is essential to consider the placement of lookout towers when building an effective detection system (Amiri et al. 2022). The landscape around a tower does not change, therefore, visibility areas of a lookout tower are fixed over time. Choosing the location for a tower is

equivalent to choosing its visibility. Towers with more favorable locations have larger directly visible areas. The range of a manned lookout tower and areas that are visible (as opposed to screened or completely blind) affect the number of fires the tower is expected to detect in a timely manner. Therefore, this characteristic of lookout towers can be thought of as inputs that are used in the production of wildfire detection.

In our production framework, towers with high visibility profile should produce superior wildfire detection performance. Visibility areas capture the different geographical attributes around a tower that can play a role in successful detection. High visibility increases the ability to provide accurate information about the location of a fire (Rego et al. 2013; Rego and Catry 2006). Visibility influences the time a wildfire takes to be detected, lookout towers with higher proportions of visible areas detect fires quicker (Sakellariou et al. 2022). Conversely, if a tower has low visibility and significant rugged or blind areas in its SR, fires can go undetected for longer which is harmful for the SR (Çoban and Bereket 2020).

We specify the area directly visible (x_1) and indirectly visible (x_2) as wildfire detection inputs. Table 4.1 summarizes our input and output variables. We develop three output-oriented bias-corrected DEA models with variable returns to scale.

Table 4.1 Input and Output Variables for Data Envelopment Analysis

Variables	Unit	Definition
Fires detected (y_1)	Fraction	Proportion of fires detected by a lookout tower
Early detection time (y_2)	Hours	Average early detection time
Directly visible area (x_1)	km ²	Area directly visible by lookout tower
Indirectly visible area (x_2)	km ²	Area indirectly visible by lookout tower

Table 4.2 shows the combinations of outputs and inputs used in the respective models. All the models use a multi-output case by considering both detection outputs discussed above:

proportion of wildfires detected (y_1) and average early detection (y_2). Model A specifies a single input, the area directly visible (x_1). Model B includes two inputs: areas directly (x_1) and indirectly visible (x_2). Finally, model C specifies a single input x_3 that is the sum of both directly and indirectly visible areas ($x_1 + x_2$). Each model is estimated separately during the time period 2006 – 2021.

Table 4.2 DEA Models for Technical Efficiency Analysis

Model	Inputs	Outputs
A (2 outputs, 1 input)	x_1	y_1 and y_2
B (2 outputs, 2 inputs)	x_1 and x_2	y_1 and y_2
C (2 outputs, 1 input)	$x_3 = x_1 + x_2$	y_1 and y_2

Chapter 5. Technical Efficiency Estimates for Wildfire Detection

Table 5.1 offers summery statistics of the bias-corrected technical efficiency estimates of our models, pooled across all years. As the results across the three models are similar, we proceed to interpret model A which is the most parsimonious model. Mean output technical efficiency is estimated to be approximately 1.059. That is, on average, lookout towers must increase their detection output by 5.9% in order to become technically efficient. In other words, our results show that lookout towers are very close to the technical efficiency frontier and are performing well. There is very little room for towers to increase the production of detection outputs. The results also show that the distribution of technical efficiency is significantly skewed to the right. Many units have scores close to one such that the median technical efficiency is 1.005.

Table 5.1 Bias-corrected Technical Efficiency Scores Pooled across All Years

Model	25th quantile	Median	Mean	75th quantile
A	1.001	1.005	1.059	1.032
B	1.001	1.006	1.057	1.032
C	1.001	1.005	1.058	1.031

The scatter plot in Figure 5.1 illustrates bias-corrected technical efficiency measures for all lookout towers each year. To complement the scatter plot and understand the change in technical efficiency over time, a linear trend represented by the solid red line is included. The time trend line shows that the bias-corrected technical efficiency estimates are decreasing over time and approaching closer to a value of 1. The technical efficiency scores are reducing; therefore, lookout towers are becoming technically efficient in carrying out detection. For a more focused insight, yearly distributions of lookout tower technical efficiency scores are included in Appendix E.

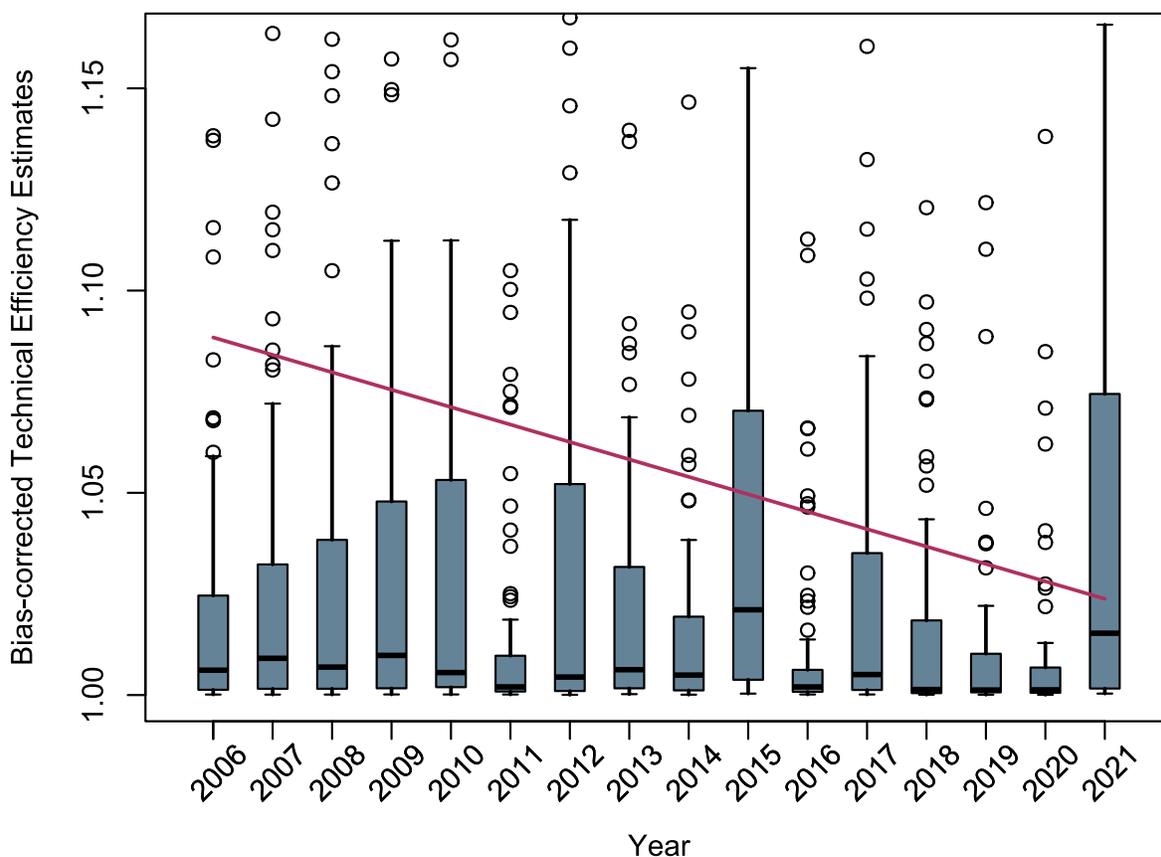


Figure 5.1 Bias-corrected Technical Efficiency Scores Over Time

The bias-correction procedure relies on resampling and averaging techniques; therefore, in finite samples, it derives individual tower technical efficiency scores that are greater than one. Note that this is not a weakness of the estimator as the DEA bias-correction approximates the true DEA score on average. However, it is still useful to have criteria to identify units that are at the production frontier (or arbitrarily close to it). Such a criterion is necessary because the true technical efficiency frontier is not explicitly identified in the bias-correction method. Therefore, to identify units at the technical efficiency frontier, we use the criterion that units with bias-corrected scores less than or equal to $1 + \varepsilon$ are technically efficient. We use five different criteria: $\varepsilon \in \{0.005, 0.01, 0.015, 0.02, 0.05\}$. For example, if we use $\varepsilon = 0.01$, towers with bias-corrected technical efficiency score $\theta \leq 1.01$ are considered to be technically efficient. Note that a score of

1.01 means that, given the visibility profile of the tower (i.e., holding constant), the lookout tower must increase detection output by 1% to be at the theoretical efficiency frontier. Therefore, for $\varepsilon = 0.01$, towers that need to improve detection output by 1% or less are classified as technically efficient towers, and towers that need to improve outputs by more than 1% are not at the technical efficiency frontier.

Table 5.2 shows the number of towers, the number of fires and the proportion of technically efficient lookout towers for the various criteria, by year. As expected, the proportion of technically efficient towers increases as we use less stringent criteria. For the remainder of our study, we use 1.01 as the threshold for classifying if a lookout tower is technically efficient or not.

Table 5.2 Proportion of Technically Efficient Lookout Towers Based on Different Thresholds

Year	N	Average θ	Number of fires	% effi. (1.005)	% effi. (1.01)	% effi. (1.015)	% effi. (1.02)	% effi. (1.05)
2006	112	1.020	1,938	46.429	55.357	66.964	72.321	87.500
2007	104	1.210	1,337	42.308	50.962	60.577	67.308	82.692
2008	104	1.039	1,722	46.154	58.654	61.538	66.346	79.808
2009	107	1.073	1,673	39.252	51.402	58.879	63.551	76.636
2010	106	1.086	1,892	48.113	57.547	59.434	59.434	73.585
2011	81	1.030	1,173	60.494	75.309	75.309	79.012	86.420
2012	92	1.109	1,605	52.174	60.870	65.217	66.304	73.913
2013	83	1.049	1,218	46.988	59.036	62.651	67.470	83.133
2014	94	1.028	1,451	50.000	62.766	70.213	75.532	88.298
2015	107	1.060	1,857	27.103	38.318	45.794	47.664	72.897
2016	92	1.017	1,436	69.565	78.261	83.696	84.783	92.391
2017	88	1.040	1,265	48.864	62.500	64.773	69.318	81.818
2018	88	1.029	1,357	60.227	67.045	73.864	76.136	84.091
2019	73	1.014	1,076	64.384	73.973	79.452	84.932	94.521
2020	55	1.015	750	70.909	78.182	81.818	81.818	90.909
2021	72	1.078	1,424	33.333	45.833	50.000	51.389	68.056
Total	1,458	1.059	23,174	49.314	59.945	65.432	68.861	81.893

Figure 5.2 plots the proportion of technically efficient and not efficient towers for each year ($\varepsilon = 0.01$ henceforward). The figure offers a visual depiction of column (%effi. 1.01) from Table 5.2. The blue area represents the proportion of technically efficient towers whereas red represents the proportion of lookout towers classified as not technically efficient. The percentage of lookout towers that are classified as technically efficient is increasing over time. Note that for all the years, except 2015 and 2021, there are more technically efficient towers (proportion more than 50%) than not technically efficient.

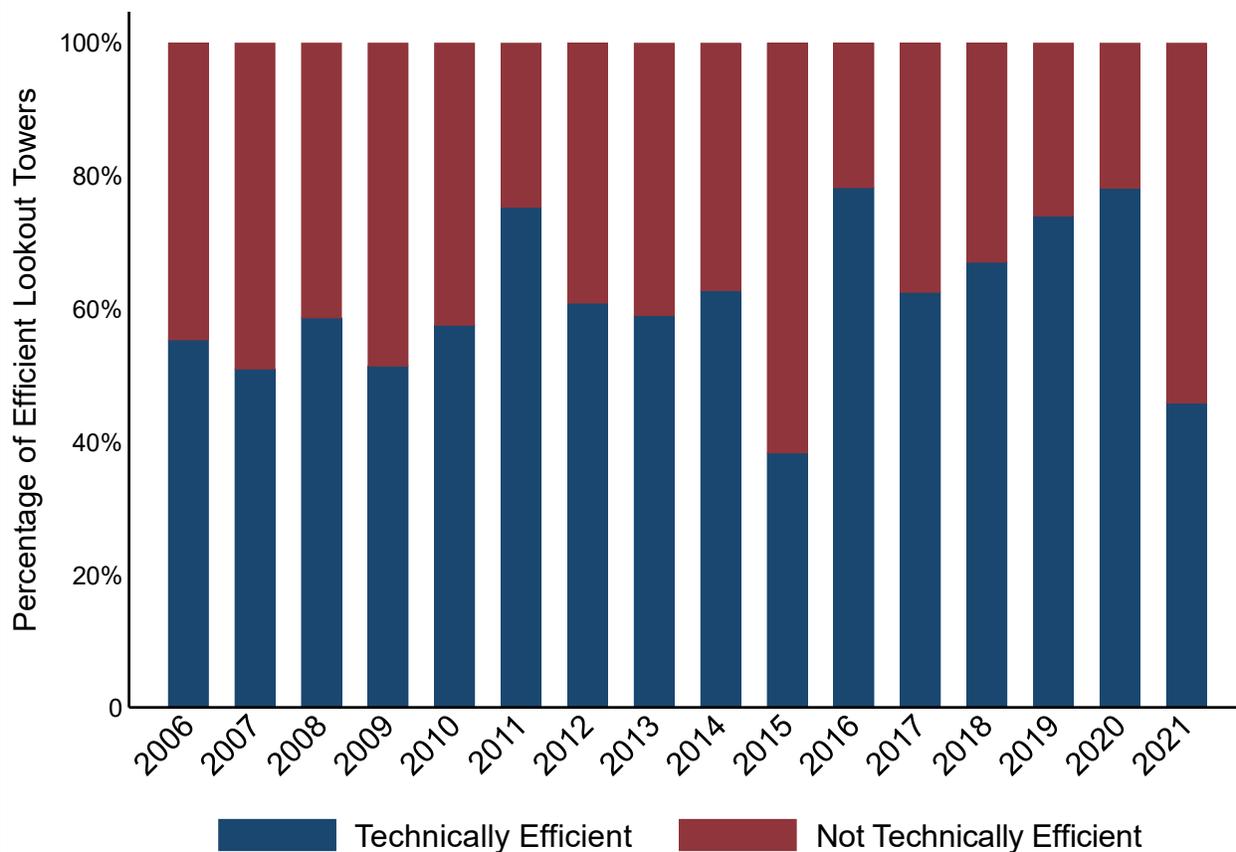


Figure 5.2 Classification of Lookout Towers

To further understand the relationship of the number of active towers and the number of fires on the proportion of lookout towers, we develop two simple regression models. Specifically, we estimate the following equation:

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t,$$

where Y_t is the proportion of technically efficient lookout towers in year t and X_t is:

Model (1): number of active towers in year, t , and

Model (2): number of fires in year, t .

	(1)	(2)
β_0	99.015*** (14.970)	94.574*** (10.778)
β_1	- 0.417** (0.162)	- 0.023*** (0.007)
N	16	16
R ²	0.321	0.420

Standard errors are in parenthesis. * p<0.10, ** p<0.05, *** p<0.01

The results of the two OLS regressions are presented in Table 5.3. For the first model, the number of towers in a year is negatively associated with the proportion of technically efficient lookout towers. The addition of one lookout tower is correlated with a reduction of 0.417 percentage points (out of 100) in the proportion of technically efficient lookout towers in a year. In other words, the addition of two lookout towers is associated with a decrease in the proportion of technically efficient towers by almost one percentage point. On the other hand, for model (2), the addition of 40 wildfires in a year is associated with a reduction of the proportion of technically efficient lookout towers by approximately 1 percentage point.

We now shift our focus to the spatial distribution of technical efficiency. Figure 5.3 shows a heat map of technical efficiencies across the FPA. For each of the 127 towers we compute the average technical efficiency score from 2006-2021, where the lower score bound (most technically efficient lookout average) is 1.001, the upper score bound (least technically efficient lookout average) is 3.110, and the median score is 1.026. Shades of deeper green represent higher technical efficiency (lower technical efficiency scores), while weaker shades of green represent lower technical efficiency (higher efficiency scores).

In the next chapter, we explore non-discretionary drivers of technical efficiency. The goal is to develop machine learning models to perform binary classifications of lookout towers. Specifically, we are interested in examining whether local weather information can be used to determine if a tower is at the technical efficiency frontier or not.

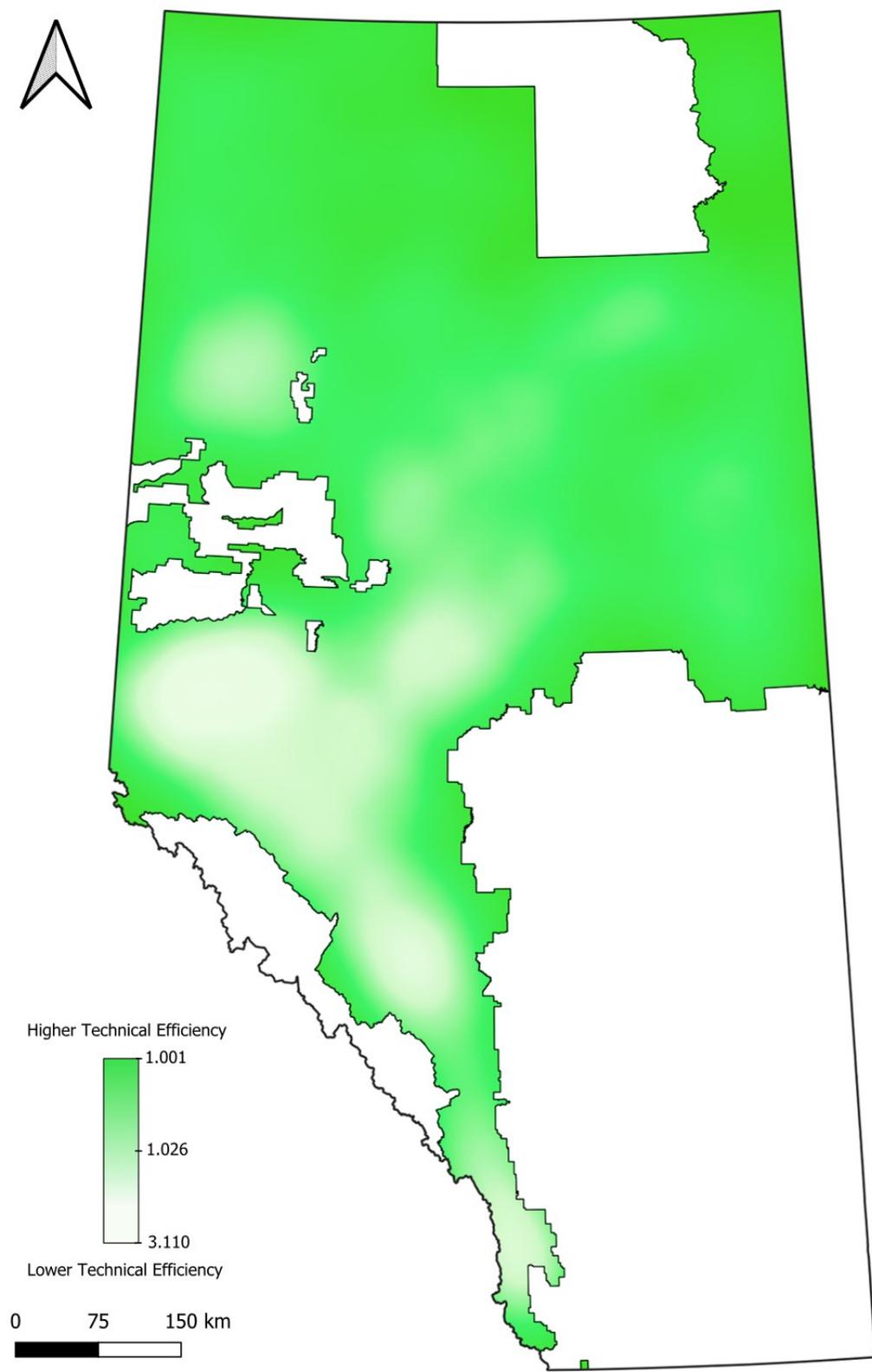


Figure 5.3 Heat Map of Average Technical Efficiency Scores

Chapter 6. Machine Learning to Predict Technical Efficiency

The statistical tools offered by Machine Learning (ML) are widely used because of their ability to analyze complex datasets and make accurate predictions (Reichstein et al. 2019). This chapter applies machine learning to predict the technical efficiency of lookout towers. Specifically, we investigate the influence of weather conditions on a lookout tower's ability to effectively convert visibility into successful wildfire detection. To study this, we train and tune a variety of classification models using weather variables as predictive features. We then evaluate the predictive performance of the models (section 6.4) and the feature importance of the weather variables (section 6.5).

ML is prevalent in the field of wildfire research. Papers in this literature often focus on using ML techniques to predict wildfire risk and use imaging algorithms to enhance wildfire detection. With regards to wildfire risk, many papers examine the relationship between weather and wildfire. Sakr et al. (2010) present a wildfire risk prediction model using Support Vector Machines (SVMs), accounting for weather conditions to predict wildfire hazard levels. Their study demonstrates the ability of ML models using high-frequency data on daily number of fires to accurately predict daily risk of wildfire occurrence.

Janabi, Shourbaji, and Salman (2018) compare SVMs against a variety of Neural Network (NN) techniques for predicting forest fires. They develop four different neural network models: Cascade Correlation Network, Multilayer Perceptron Neural Network, Polynomial Neural Network, and Radial Basis Function. They find that SVM outperforms NN, and it is therefore more suitable for the task of predicting forest fires.

Stojanova et al. (2012) use predictive models such as K-Nearest Neighbors, Naïve Bayes, Bayesian Networks, SVMs, and Random Forests (RF) to estimate the risk of wildfire occurrence. Their models incorporate GIS data, remote sensing images, and weather forecast models to make predictions. They conclude that combinations of many Decision Tree (DT) models produce the highest accuracy in predicting the probability of a fire occurring at a specific location at a specific time.

Xie et al. (2022) use SVM, RF, and XGBoost to study wildfire risk and compute feature importance values (feature importance is discussed in detail in section 6.5). Using data on wildfire triggering factors based on topography, human activities, and meteorology, the study finds that eXtreme Gradient Boosting (XGBoost) models exhibit better performance. They find that precipitation, air temperature, and land cover show significant effects when predicting wildfire occurrence and behavior.

Dong et al. (2022) explore the effects of geographical and temporal variations on monthly time series of wildfires. They develop ML models such as XGBoost, RF, SVM, and DT. They find that XGBoost outperforms other models and that weather variations over time have a significant influence on the time series and spatial dispersion of wildfire occurrences.

Collins et al. (2018) and Gibson et al. (2020) assess the performance of RF classifiers to map wildfire severity. The studies use satellite-based wildfire maps by using landscape imagery and biophysical characteristics, such as ground cover, vegetation, and soil. Collins et al. (2018) find that RF classifiers complemented with spectral data provide a reliable method for mapping fire severity across heterogeneous landscapes. Gibson et al. (2020) conclude that the model performs the best for cases when full forest crown scorch occurs.

Moreover, the literature has shown that the incorporation of imaging information can improve the performance of wildfire detection models. Zhang et al. (2018) develop a Region-based Convolutional NN (R-CNN) model for the purpose of detecting wildfire smoke from photos. Their study relies on simulated (synthetic) smoke imagery of two kinds: photos of actual smoke pasted on a forest background and smoke generated from a rendering software which is then inserted on a forest background. The study concludes that this type of NN model can identify and localize smoke from videos.

Barmpoutis et al. (2019) train a R-CNN model for detecting wildfires using annotated terrestrial-based images. Their experiment aims to test the ML model using actual fire images and images that contain objects that are fire colored. The authors find that their model results in high true positives while reducing false positives from fire-colored objects. This can help in successfully determining wildfires using deep learning techniques.

Dutta, Das, and Aryal (2016) examine the relationship between climate data and fire incidence. The study relies on NASA Active Fire and Burned Area satellite imagery and weather data. They demonstrate high accuracy in predicting hot-spots and correctly identifying bush-fire incidents in Australia. The authors believe predictive systems and statistical learning help understand climatic variations on bushfires within a weekly temporal scale.

Zhao et al. (2018) construct SVM, Artificial Neural Network and CNN models to optimize wildfire detection. Their models rely on UAV imagery of wildfire and wildfire smoke to develop saliency methods for detection. The methods used in this study efficiently locate core fire regions and even very small ignition zones from the aerial images.

6.1. Weather Data as Machine Learning Predictors

Severe fire seasons have been linked with higher air temperatures, lower humidity, variability in wind patterns, and rain-scarce regions (Jain et al. 2022). In fact, these variables are part of the most widely used wildfire weather index, the Canadian Fire Weather Index System (FWI). It is therefore important to account for weather in developing wildfire behavior and detection models in both operational and research contexts (Wagner 1987). In this study we use six meteorological variables as ML predictors: air temperature, humidity, incoming radiation, snow water, precipitation, and wind speed.

Air temperature and relative humidity are two important factors that influence wildfire incidence (Abatzoglou et al., 2019). Higher maximum temperatures are favorable for wildfires (Walker et al. 2019; Halofsky, Peterson and Harvey 2020; Canepa and Drogo 2021). Low levels of humidity accelerate desiccation of biomass fuels, making them more combustible (Herawati and Santoso 2011). The term ‘crossover’ is widely used in the forestry industry in Canada, it explains a simple guideline for the threshold conditions of extreme fire behavior: when relative humidity is equal or lower than the air temperature (Lawson and Armitage 2008; Cruz and Alexander 2019). These two weather conditions working in opposite directions increase wildfire risk and can result in severe fire seasons (Jain et al., 2022). The 30-30-30 rule is another widely used rule of thumb for defining extreme fire behavior: temperature greater than 30 degrees Celsius, relative humidity 30% or lower, and winds of 30 kilometers per hour or more lead to fires that are difficult to control until weather conditions change (English, 2018; Mahdavi, 2023). Thus, in conditions favoring wildfire incidence and spread, lookout towers must maximize timely detection to perform efficiently to mitigate damages.

Incoming radiation is a significant weather variable to consider when predicting if a lookout tower is technically efficient in detecting wildfires. There is a relationship between net radiation and soil moisture (Eltahir, 1998). Incoming radiation explains aridity which is an indicator of soil moisture and humidity (O et al., 2020). Larger wildfires tend to occur when fuel moisture is low which promotes easier fire ignition especially when temperatures are high, and humidity is low (conditions that reduce fuel moisture and increase fire spread risk) (Krueger et al., 2015). Low fuel moisture leads to vegetation becoming more flammable, increasing severity of wildfires (Sharma et al., 2020). Therefore, we account for incoming radiation when exploring the association between weather and technical efficiency of wildfire detection.

Wind speed, rainfall, and snow water equivalent affect wildfire risk and incidence. Higher recorded wind speeds are linked to lower fuel moisture which can influence the rate of spread and intensities of wildfires (Banerjee et al. 2020). Additionally, wind is one of the most important variables in the spread of a wildfire, making it a useful factor to consider (Lookout Observer Manual 2022, p.164). Lower precipitation promotes wildfire occurrence (Fang et al. 2021). Moreover, larger snow water equivalent values are linked to a smaller likelihood of wildfire severity (McGrath et al. 2023). Considering these studies, we account for wind speed, rainfall, and snow water around the vicinity of a lookout tower and explore if they affect technical efficiency.

These findings support the significance of weather variables as predictors in machine learning algorithms. Table 6.1 summarizes the weather variables used in the machine learning algorithms, their units of measure, and a short description of what they measure.

Meteorological data used as predictors in our machine learning models are provided by ACIS at the township level. While some towers have equipment to measure local weather, many

do not. To accurately reflect local weather, the weather station observations are aggregated to lookout tower-level by applying an inverse distance-based mean with a threshold of 150 kilometers. Therefore, observations closer to the tower are given more weight in measuring weather than observations further away. The final step in preparing weather data is to aggregate monthly to yearly observations. We choose the months of April, May, June, and July to define the average of weather variables during the fire season. These months capture 75.29% of wildfires from our dataset.

Table 6.1 Weather Variables as Predictors in Machine Learning Models

Variable	Definition
Maximum air temperature (°C)	The maximum recorded air temperature measured in degrees Celsius.
Humidity (%)	Relative humidity is a measure of the relative amount of moisture present in the air at a specific temperature recorded 2 meters above the ground. It is expressed as a percentage indicating how close the air is to being saturated with moisture (100% relative humidity).
Incoming modelled radiation (MJ m ⁻²)	The amount of solar energy received per meter square estimated using daily maximum and minimum temperatures, date, station latitude and elevation, and other variables. Please refer to Appendix H for more information about the methodology used by ACIS to estimate incoming radiation. Measured in mega joules per meter square.
Snow water equivalent (mm)	The amount of water that would be obtained by melting the depth of snow. Measured in millimeters.
Precipitation (mm)	The amount of rainfall recorded in millimeters.
Wind speed (kmh ⁻¹)	Speed of wind measured at a height of 10 meters above the ground, a standard height for weather stations. Measured in kilometers per hour.

6.2. Machine Learning Algorithms

Machine learning models can be categorized into two broad groups: regression models and classification models. Regression models are applied when the outcome or target variable is a continuous numerical value, for instance, predicting and estimating house prices (James et al. 2013). On the other hand, problems that aim to assign output predictions to finite numbers or discrete categories are categorized as classification problems in machine learning (Bishop 2006). Therefore, classification models are preferred for predicting whether lookout towers are technically efficient based on weather variables. Technical efficiency is the target variable that is to be predicted. It is a binary variable with two levels, 0 indicating technical inefficiency and 1 indicating that the lookout tower is technically efficient. Lookout towers with a technical efficiency score of 1.01 or less are classified as technically efficient while those with higher scores are said to be not technically efficient.

The predictors used are the weather-related variables discussed in the previous section. Nine classification models are constructed: Regularized Logistic Regression, Naïve Bayes, K-Nearest Neighbor, Support Vector Machines, Neural Networks, Decision Tree, Random Forest, Adaptive Boosting, and eXtreme Gradient Boosting. The inclusion of these nine classification models represents a variety of ML approaches. Therefore, evaluating these models offers a well-rounded assessment of predicting the technical efficiency class of lookout towers based on weather variables.

Logistic Regression aims to estimate the probability that the target variable belongs to the positive (lookout is technically efficient) or negative (lookout tower is not technically efficient)

class (James et al. 2013). The model estimates coefficients by maximizing the likelihood of the instance being in a given class in the training data. If the probability of an instance falling in the positive class is (0.50) 50% or more, the data point is classified into the said class (Géron 2019). On the other hand, an observation with a probability of the instance less than 50%, is classified into the negative class. In our application of the logistic regression model, we use the boosted logistic regression model. ‘Boosting’ in machine learning is a technique that aims to improve the performance of a model by combining weak learning algorithms into a stronger ‘ensemble’⁴. Ensembles train a series of models with the focus of correcting the errors in the predecessor models and sequentially improving overall predictability. Thus, we obtain a final model that progressively becomes more accurate (Hastie, Tibshirani and Friedman 2001; Géron 2019). In the context of logistic regression, boosting combines individual logistic regression models with each subsequent model focusing on rectifying the errors of its predecessor. As a result, enhancing the prediction capability of the final model.

K-Nearest Neighbor (KNN) classification models attempt to estimate the conditional distribution of the target variable given a set of predictors based on neighboring instances of a given observation. The algorithm identifies K neighbors in the training data and estimates the classification of the lookout tower based on the majority class of the nearest neighbors to the lookout tower of interest (James et al. 2013; Bishop 2006). While this algorithm helps classify the target variable into the respective classes and does not impose assumptions on the distribution of the training data, predictions are sensitive to the choice of parameter K (number of neighbors). For small values of neighbors, the model tends to overfit. As this parameter gets larger the decision

⁴ ‘Ensemble’ in ML is a technique that combines predictions and outputs of multiple individual models. The aim is to improve the overall performance by aggregating results of various models. As a result, the predictions collected are more robust and accurate as compared to any one individual model.

boundary becomes strict, tends closer to a linear boundary, and leads to a low-variance (the model is stable) but high-bias (model simplifies the patterns in the data) classification (Hastie et al. 2001; James et al. 2013).

Support Vector Machines (SVM) split the data by constructing a decision boundary often called a hyperplane that maximizes the separation of the different classes. In a \mathbb{R}^n space, the hyperplane is of dimension \mathbb{R}^{n-1} . For instance, in a two-dimensional space, the hyperplane is a line. Using the training dataset, the algorithm decides an appropriate hyperplane by implementing support vectors. A support vector is a data point from the training data that lies closest to the hyperplane. These support vectors play a crucial role in defining an appropriate hyperplane that attempts to neatly dissect the two classes (positive and negative instances) (Bishop 2006). The model attempts to maximize the margin between the two classes by finding the best line or surface that separates the data while correctly classifying as many instances as possible. SVMs are good for complex classification when the data is small or medium-sized (Géron 2019) and can be useful for binary classification tasks in which there are two classes (James et al. 2013). One drawback of SVM models is that they may have difficulties defining hyperplanes if the dataset has overlapping or imbalanced class distributions, requiring fine tuning and modifications to the algorithm.

Decision Tree algorithms follow a hierarchical model for learning (Alpaydin 2020). They consist of constructing trees by using a sequence of recursive splits. A Decision Tree is a structure resembling a flowchart with nodes representing an attribute (or feature). The tree begins with a root node by randomly choosing one predictor. Then splits are made with branches stemming out of the root node. The branches lead to internal nodes and are based on a decision rule (a logical argument or inequality). The final node is called the leaf node (terminal node) which rules the prediction or decision. In our case, the terminal nodes take up binary values of 1 for technically

efficient or 0 for not technically efficient. Recursive binary splitting (repeatedly dividing the dataset into homogenous sets based on criteria) eventually grows a classification tree. Thus, the feature space partition is captured within a single tree. The predictions are then made based on the predictors' values in the unseen testing data (James et al. 2013; Hastie et al. 2001).

The Random Forest approach refers to an ensemble of Decision Trees. DTs by themselves are noisy but benefit from averaging since they capture complex interaction between a set of predictors (Hastie et al. 2001). The Random Forest algorithm begins by drawing bootstrap samples from the data. For each sample, it grows a tree. The predictions of individual trees are collected and then the target variable is assigned its predicted class based on the class that gets the most votes (Liaw and Wiener 2002; Géron 2019).

Neural Networks consist of layers of connected neurons that work together to find patterns in the dataset to make final predictions (Hastie et al. 2001). Goodfellow et al. (2017) and Géron (2019) discuss the workings of a neural network model. Neurons are the first component of a neural network. They are basic processing units that transform a given input instance into meaningful information for the neural network to learn from. Neurons also calculate the weighted sum of inputs which assigns an importance to each input. A component for the bias is also accounted for, allowing for finer tuning. This helps the network understand complex patterns. The information from each neuron is passed to an activation function. This function allows the neural network to learn complex relationships between different classes and features. The activation function can be thought of as a switch for each neuron. The output of the activation function acts as the input for the neuron in the next layer. This function produces an output between 0 and 1 which signals how much activation is given to each neuron. Higher activation informs the network about how strongly the given neuron influences the learning of patterns in the training dataset.

This process is repeated till the final output is achieved in the last layer of the neural network. The weights and biases are adjusted based on the errors of the entire learning process. The goal is to reduce misclassification and improve the network's learnability. The neural network model used in this study uses Principal Component Analysis (PCA) as a pre-processing step. PCA is a technique used to reduce the number of dimensions and therefore, the complexity of the dataset (Suleiman, Tigh and Quinn 2016). It transforms the original inputs into principal components or uncorrelated features. One of the fundamental properties of the PCA is that it de-correlates variables in the data. These principal components are linear combinations of the original inputs and are sorted in order of variance. The components with higher relative variance capture variability in the data. Similarly, in a PCA Neural Network, the PCA approach simplifies data into components by reducing the number of dimensions. These simplified components act as predictors in the neural network. The aim of this approach is to concentrate the neural network's learning towards the components that contribute more towards the variability in the data.

Naïve Bayes is based on a probabilistic approach that relies on Bayes' theorem to calculate the probability of instances belonging in either class. Similarly, Bayes' theorem applied in an ML model finds the chances of an instance belonging to a particular class based on its predictor values. Naïve Bayes' biggest assumption is feature independence. The presence of a feature in a class is independent of other features. In the training phase, the algorithm calculates probabilities, then uses unseen data to select the class with the highest probability as the final predicted class (Hastie et al. 2001; Alpaydin 2020) .

An Adaptive Boosting (AdaBoost) classifier model starts by picking a simple model such as a Decision Tree. The model is trained, and predictions are made on the training set. Based on the performance of the model, the relative weights of misclassified observations are increased. As

such, the ensemble learns from itself. Misclassified instances with higher weights need to be rectified in order to improve the prediction of the ensemble. As a result, the subsequent model focuses more on these data points that were previously classified incorrectly. A second model is trained, predictions are made, and the weights assigned to instances are updated accordingly. This process is repeated many times. The intuition behind increasing weights is as follows: if the first model predicts many wrong instances, the weights are boosted. Therefore, the second classifier model performs better on the wrong instances. Adaptive boosting adds predictor variables gradually to make the final model more accurate (Géron 2019). After all the predictors are included in the training phase, each of them is assigned different weights based on the accuracy of the weighted training set. Finally, predictions are made using the bagging approach. ‘Bagging’ is short for bootstrapped aggregation. Bagging is a technique in machine learning that trains many models on subsets of data. The bootstrapped predictions are aggregated which reduces variance and improves the stability of the ensemble (Alpaydin 2020).

Extreme Gradient Boosting (XGBoost) is the last algorithm we choose in this study. It is a boosting algorithm like AdaBoost that combines many models to create a robust final model by sequentially adding predictors to a learning ensemble, correcting its predecessor model along the way. It generates a combination of models that attempt to improve on the errors made by the previous model. The difference between extreme gradient boosting and adaptive boosting is that the XGBoost fits the new predictor to the residual errors made by the previous predictor whereas adaptive boosting adjusts the weights at every iteration. (Géron 2019). This algorithm is useful when the data has a significant amount of noise (Friedman 2001). While XGBoost can handle large datasets, it is computationally costly and contains many hyperparameters that require tuning to obtain the best performing model.

6.3 Model Training and Hyperparameters

To achieve optimal predictive performance, machine learning models require training and fine-tuning of hyperparameter. Typically, a trained model relies on a subset of the data to learn and build the model. As such, data is divided into two parts: the training sample and the testing sample. The training set in our analysis is 80% of the original dataset (with 20% in the testing dataset). Once training models are built, we apply them to predict the class (“technically efficient” or not) of the lookout towers. We use classification models where outcomes are measured in terms of discrete variables, often referred to as class labels, e.g., towers are either technically efficient or not technically efficient.

The method of cross-validation (CV) is applied to train our models. CV is a data resampling approach that aims to prevent overfitting and estimate true prediction errors (Berrar 2018). It is a training strategy that involves randomly dividing the set of observations into k -folds (or subsamples) of approximately equal sizes. The procedure involves k iterations. We have 1,458 tower-year observations in our entire dataset. Therefore, a 10-fold cross-validation will randomly divide the training data, which is approximately 1,166 observations (80% of 1,458), into 10 sub-groups of approximately 116 – 117 observations. In the first iteration, one of the 10 sub-groups is set as the validation set and the model is trained on the remaining 9 ($k - 1$) folds. Once the model is trained it is used to predict the class label on the validation set (the sub-group that is held out). The procedure is repeated k times till all the folds (or sub-groups) are held as the validation set in different iterations. Therefore, CV predicts labels using multiple unseen testing data subsamples.

In each iteration of the cross-validation method, the model's predictive performance is calculated and stored. Usually, for regression machine learning models, the Mean Squared Error (MSE) statistic is used as the metric of performance. MSE measures the average squared difference between predicted values and actual values. A lower value indicating the model's predictions are close to the actual values, as such, offering an appropriate estimate of performance. In a 10-fold CV, there are 10 different MSE values obtained for each iteration. The final model performance measures when using cross-validation are computed as the average of each iteration's value (James et al. 2013; Rodríguez, Pérez and Lozano 2010).

For classification models, MSE is not the best measure of performance therefore, measures such as sensitivity, specificity, and Area Under the Receiver Operating Characteristic curve (AUROC) are used (Kuhn and Max 2008). The performance metric of AUROC is discussed in detail in the next section (6.4 Model Assessment). We apply CV using the Receiver Operating Characteristic (ROC) curve to measure the performance for our models. The AUROC value for each iteration is averaged out. The formula is as follows:

$$CV_k = \frac{1}{k} \sum_{i=1}^k AUROC_i \quad (6)$$

Where:

$$\begin{aligned} CV_k &= \text{k-fold cross-validation} \\ AUROC_i &= \text{AUROC value for i-th iteration} \end{aligned}$$

Cross-validation allows for efficient use of limited data by ensuring that the models are trained and tested on many different portions and subsamples of the dataset. Additionally, this reduces chances of overfitting and provides better estimates about the model's generalization

performance (how well the model performs on unseen data that was not incorporated in the training stage). Cross-validation is essential for hyperparameter tuning (Hastie et al. 2001). It helps inform the best model parameters after evaluating outputs across different combinations of parameters.

Predictions from machine learning models depend on a particular model specification, that is, a choice of the levels of the models' parameters. For example, in Decision Tree, one must specify how many splits the model should perform, or, in a Neural Network, how many layers to include in the model. These machine learning parameters, for example, 'splitrule' (for Random Forests) and 'maxdepth' (for XGBoost), are often referred to as hyperparameters. Hyperparameters influence the training of algorithms and have a significant influence on model performance (Wu et al. 2019). Therefore, the parameters of any given machine learning model need to be tuned for optimal results. One way of tuning parameters is using a random search algorithm. Random search trains models by randomly picking a combination of hyperparameter values. The approach then compares the results from using different combinations of hyperparameters based on performance metrics and chooses the specific set of parameters values that leads to optimal model performance (Wu et al. 2019; Bergstra, Ca and Ca 2012). The application of random search in this study is as such; the model picks a random combination of hyperparameter values. Then, after training, the model predicts the classes on testing data and saves the output metrics along with the hyperparameter values. This process is repeated 500 times in which the program randomly selects different hyperparameter values each time. As a result, we obtain a compilation of various hyperparameter values and the respective model output metrics.

Our goal is to select the set of hyperparameter values that generates the best performing models. Using the complied dataset, we use linear splines to find specific parameter values that maximize the AUROC value. Certain models have parameter values that are strings and not

numeric. For example, in the K-Nearest Neighbor model, the parameter ‘kernel’ can take string values such as ‘biweight’, ‘gaussian’, or ‘rectangular’. In this case, dummy variables are generated for each possible value when the hyperparameter is a categorical variable and splines are created using each one of these dummy variables. Hyperparameters are divided into percentile intervals of 20, 40, 60, and 80. Then splines are used to model AUROC as a function of the hyperparameters. Finally, the values that maximize the smoothed AUROC are the tuned parameters chosen for our machine learning models. A list of the tuned model-specific hyperparameters and their definitions is presented in appendix I.

6.4. Model Assessment

It is important to examine the performance of the different machine learning models by evaluating their ability to make predictions on new and unseen data. There are many statistics that can be used to measure a model’s predictive performance. These measures give a sense of how powerful the machine learning model is in predicting whether a lookout tower is technically efficient or not.

The most basic output metric is the confusion matrix. Figure 6.1 shows a general 2-by-2 confusion matrix used in a two-class problem. It is a tool that allows us to see frequencies of what the machine learning model predicts (prediction set) against the actual data (reference set). True positives (TP) are instances when the machine learning model correctly classifies a tower as technically efficient. Similarly, a true negative (TN) is when the machine learning model correctly predicts a lookout tower as not technically efficient. On the other hand, a false positive (FP) instance occurs when a lookout tower is not technically efficient in the dataset, but the ML model

predicts it to be technically efficient. Lastly, a technically efficient lookout tower that is classified as not technically efficient is a false negative (FN) instance.

		Reference set	
		Positive class	Negative class
Prediction set	Positive class	True Positive (TP)	False Positive (FP)
	Negative class	False Negative (FN)	True Negative (TN)

Figure 6.1 Confusion Matrix

Various measures of performance can be built based on the four values in the confusion matrix such as accuracy, precision, sensitivity (or recall), specificity, and F1 score. These measures are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Sensitivity (Recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{F1 score} = 2 \left(\frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \right)$$

The Area Under the Curve for Receiver Operating Characteristic (AUROC) combines many of the measures above and is a popular metric in evaluating classification models. As such, we proceed to use the AUROC as our main model performance measure. The AUROC is based on the Receiver Operating Characteristic (ROC) curve. The ROC curve plots the true positive rate, i.e., sensitivity (recall), against the false positive rate, i.e., the ratio of instances belonging to the negative class incorrectly classified as positive (Géron 2019). Note that the false positive rate is equal to one minus the true negative rate (ratio of negative instances that are correctly classified which in fact is the true negative rate or specificity). Therefore, a ROC curve plots sensitivity against one minus specificity.

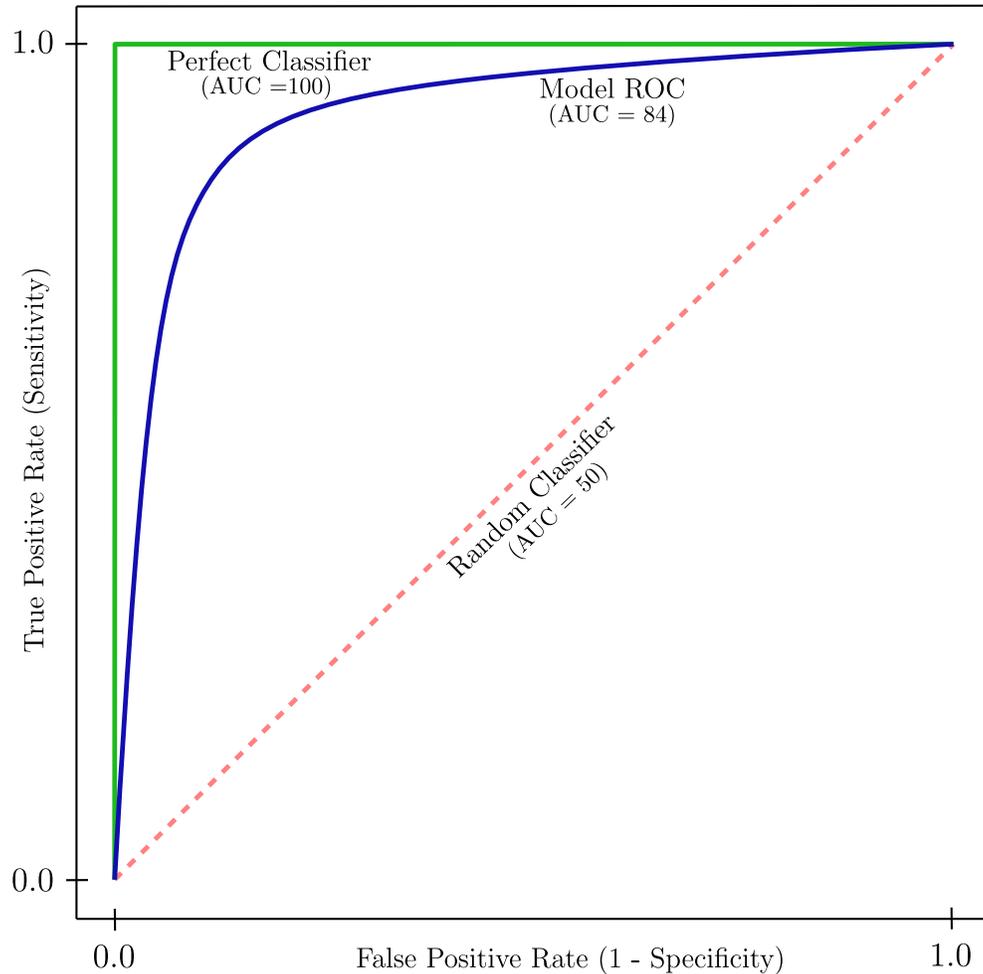


Figure 6.2 General ROC

We use Figure 6.2 as a reference to discuss the characteristics of a general ROC curve and the intuition behind AUROC values. The figure shows a ROC curve from some ML prediction represented by the blue line. The axes are the metrics of sensitivity and 1 – specificity. The dashed red line represents the ROC curve of a random classifier model that predicts instances purely based on chance. This is undesirable for any machine learning model. In the context of our study, we would interpret this ROC curve by concluding that the model’s performance is no better than randomly guessing whether a lookout tower belongs to the ‘technically efficient’ class based on local weather around it. The green line represents a perfectly predictive model, i.e., the model

predicts all instances correctly. Any good classification model's ROC curve should be as far away from dashed red line as possible and close to the green line, hugging the top-left corner of the graph.

Finally, AUROC is the area under the ROC curve. A perfect predictive model (green ROC curve) where all predictions made are correctly will have a value of 100 (Google Developers 2023). An ML model that performs no better than relying on chance will have an AUROC approaching 50 (dashed red ROC curve), such as the outcome of flipping a coin for an infinitely large dataset (Géron, 2019; James et al., 2013). In other words, a model with AUROC equal to 50 is not able to separate instances into classes accurately and randomly guesses the class that an instance will belong to. Therefore, AUROC measures the trade-off between sensitivity and specificity in a classification task. As such, it speaks towards the discrimination ability of the model.

Results

In the discussion that follows, we use the AUROC to evaluate our nine ML models. Additional model assessment measures are available in Appendix J. Figure 6.2 plots the ROC for all models. Logistic Regression (Logit) has the lowest AUROC value of 70.236 followed by Naïve Bayes (71.035), Decision Trees (79.949), and Neural Network (81.930). Models with higher AUROC values include Support Vector Machines (92.388), eXtreme Gradient Boosting (91.849), Random Forests (95.383), Adaptive Boosting (94.108), and K-Nearest Neighbors (98.407). A high value indicates that the model is accurate at classifying whether a lookout tower is technically efficient (positive instance) or not (negative instance) based on the weather around them. The curve shows

a high true positive rate while maintaining a low false positive rate. Therefore, the model performs well and correctly predicts more true positives rather than false positives, highlighting the robustness of the model.

One way to evaluate the AUROC values, is by following a rule of thumb used in the works of Adamecz-Völgyi, Henderson and Shure (2020): a model's ability to predict is considered "good" if the AUROC is more than 80 and "great" if the AUROC is more than 90. Based on the results in Figure 6.4, models that perform great are SVM, XGBoost, RF, AdaBoost, and KNN. Whereas NN is considered to be a good classifier model. On the other hand, Logit, NB, and DT have AUROC values of less than 80.

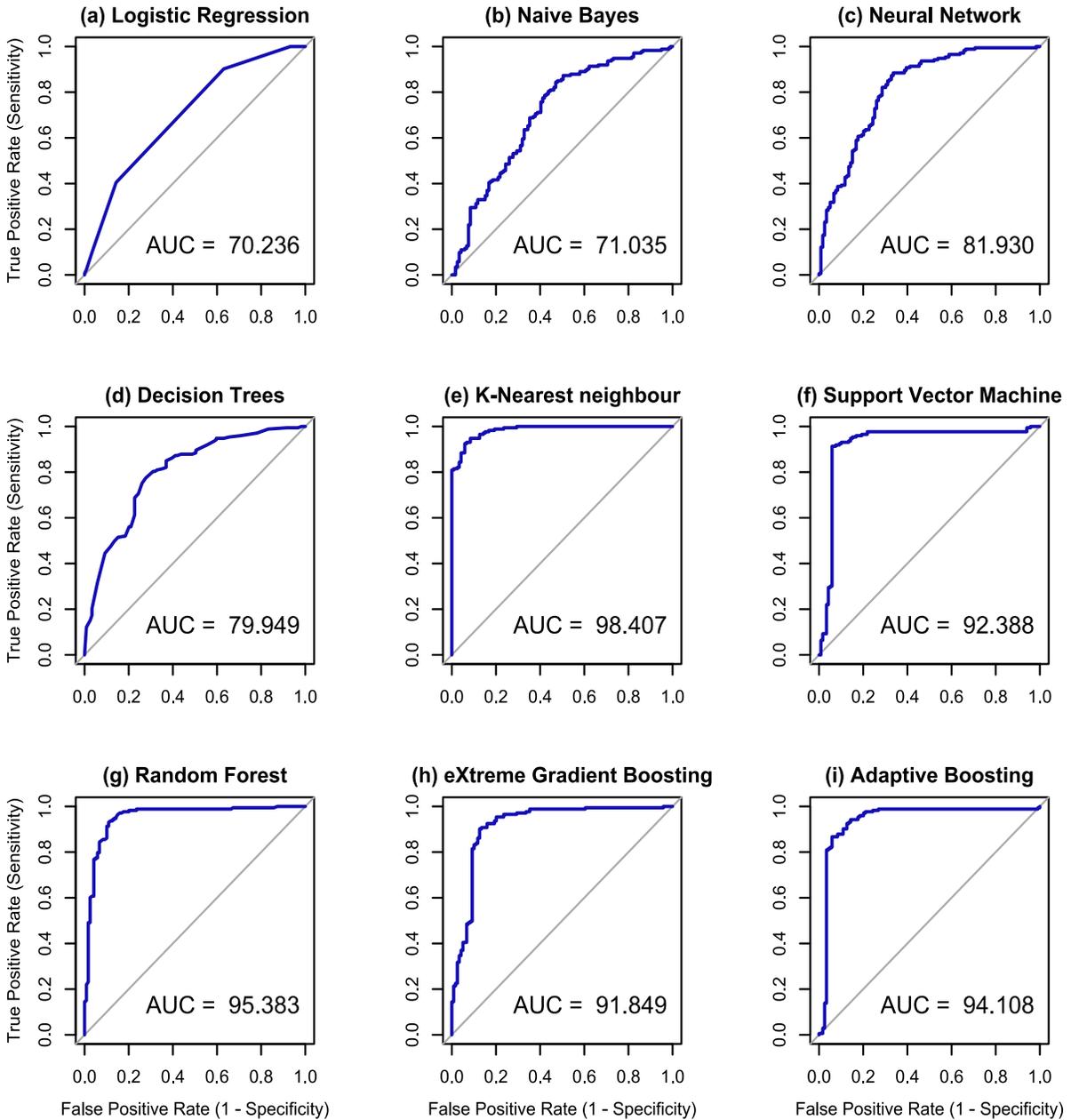


Figure 6.3 AUROC Graphs for Machine Learning Models

6.5. Feature Importance

First introduced by Breiman (2001), feature importance is a model-independent approach that can be implemented across different machine learning algorithms (Pedregosa et al., 2011). Estimating feature importance is a posterior evaluation technique that evaluates the influence of predictors in a ML model. It helps understand which variables play a significant role in our machine learning models. Feature importance does so by ranking the influence of variables (predictors) in predicting whether lookout towers are technically efficient or not. The importance of a feature, a weather variable in this study, will be quantified by the increase in the prediction error of the model when the variable is removed from the dataset entirely. The intuition is that the magnitude of the drop in model performance after removing a feature shows how influential that feature is for the model to make accurate predictions.

A specific type of feature importance is “permutation feature importance”. Instead of dropping a feature completely, like in the general feature importance approach mentioned above, permutation feature importance relies on randomly shuffling (or scrambling) a given predictor’s values while holding all other variables fixed. The model then predicts the class label, and the performance metrics are computed. Similar to traditional feature importance where a feature is completely omitted, in permutation feature importance, a feature is important if after shuffling its values, the model performance decreases significantly. The magnitude of the drop in model performance concludes how heavily the model relies on the given feature to make correct predictions. Conversely, a feature is less important if shuffling its values does not change model performance significantly (Molnar, 2020; Pedregosa et al., 2011).

Our approach for calculating permutation feature importance of weather variables follows a method discussed by Pedregosa et al. (2011). For feature j of model m , each iteration k scrambles the feature's values randomly. Then the new AUROC value (s_{kj}) of the model is collected after the model has made predictions using dataset which now has the scrambled values for feature j . The magnitude of influence or "importance" of feature j on the AUROC of the model is calculated by applying the following formula:

$$i_{jkm} = \frac{s_m - s_{kj}}{s_m} \quad (7)$$

Where s_m is the original AUROC of tuned model m prior to any scrambling of feature j . This is carried out for each of the six weather variables for each model. Finally, the feature importance values are averaged out over the k iterations (in our study we chose 1000 iterations):

$$i_{jm} = \frac{1}{K} \sum_K i_{jkm} \quad (8)$$

As such, the average permutation feature importance for each feature is computed. Figure 6.4 shows the model specific feature importance. A higher permutation feature importance value indicates that the AUROC values reduced significantly when the given feature is shuffled. The given feature influences model performance to an extent and therefore is of some importance to the model. As seen in Figure 6.4, the permutation feature importance values vary from model-to-model.

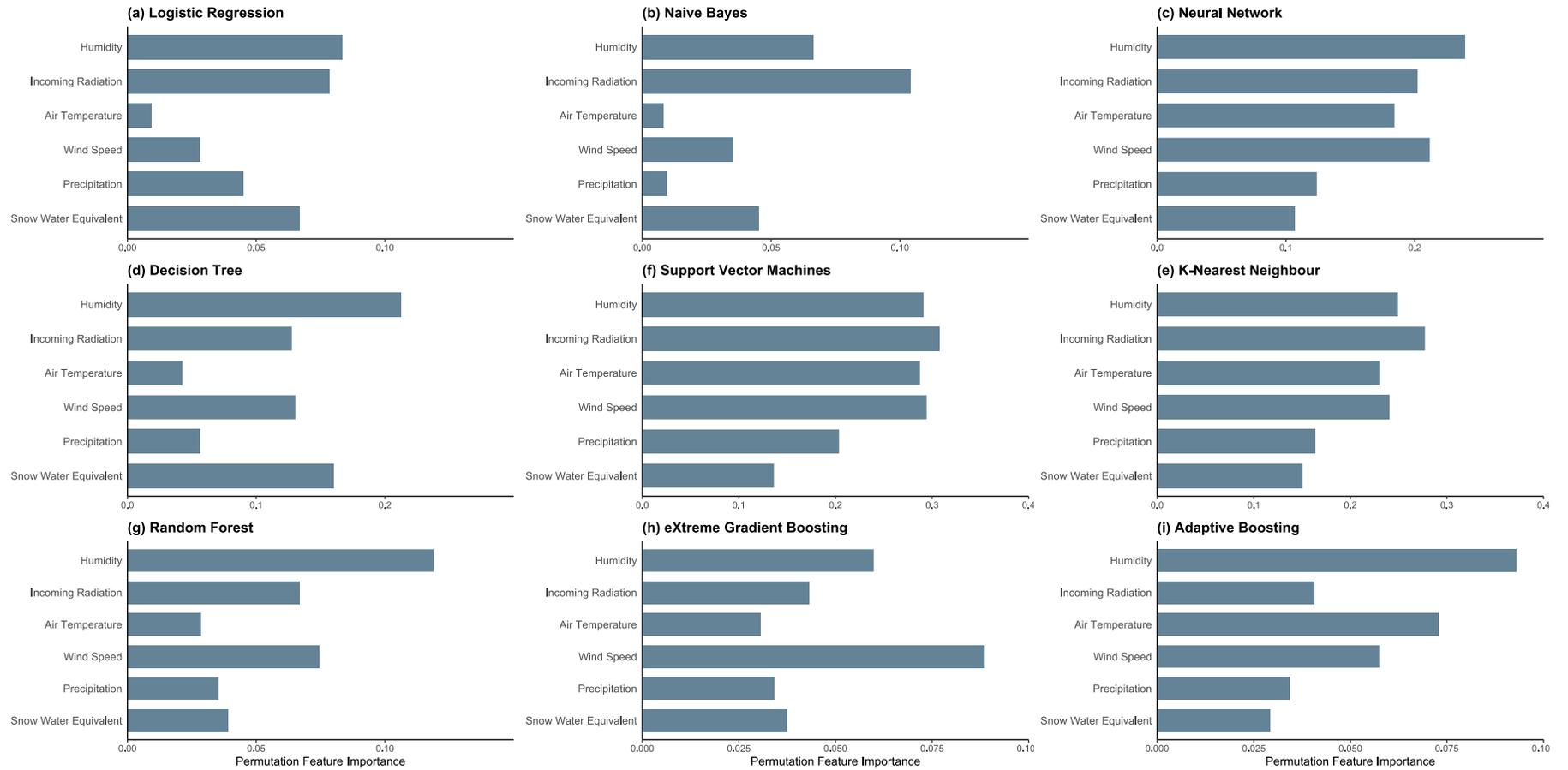


Figure 6.4 Permutation Feature Importance

To better understand the importance of weather variables on the technical efficiency of lookout towers and construct one measure of importance, we introduce a method that combines model-specific feature importance. We use the feature importance from all models but models that perform better than others are given more weight. The proposed formulation for calculating feature importance in our study is referred to as the ‘AUROC Weighted Permutation Feature Importance’. This is computed by multiplying the model-specific feature importance values by weights assigned to each one of the respective nine machine learning models. Thus, providing an aggregated feature importance value across all of our nine models.

To accurately reflect the influence of different models in the final measure, each model is assigned a weight by using an inverse distance-based approach. A perfect model predicts all instances correctly and has an AUROC value of 100. This is used to rank the different models based on their ‘distance’ from being a perfect model. As such, the AUROC Weighted Permutation Feature Importance accounts for all model performances. The following equations are used to calculate the weights (w_m). First, we calculate the inverse distance:

$$\text{Inverse distance}_m = \frac{1}{100 - \text{AUC}_m} \quad (9)$$

Where AUC_m are the values presented in Figure 6.2 for model m . Next, the weight (w_m) is obtained by dividing the inverse distance and the sum of inverse distances given by the equation:

$$w_m = \frac{\text{Inverse distance}_m}{\sum_m \text{Inverse Distance}} \quad (10)$$

Therefore, this process penalizes models that performed relatively poorly by assigning them a smaller weight. As such, models that performed relatively poorly will play only a small role in the final AUROC Weighted Permutation Feature Importance but are still accounted for. The follow equation defines our final permutation importance value:

$$\text{AUROC Weighted Permutation Feature Importance} = w_m \times i_{jm}$$

Figure 6.5 presents a bar plot of the AUROC Weighted Permutation Feature Importance. The AUROC Weighted Permutation Feature Importance for humidity is 0.189, making it the most important weather variable that predicts the ability of lookout agents to efficiently translate the visibility profile of a tower into wildfire detection. Followed by incoming radiation (0.184), wind speed (0.171), and air temperature (0.151). Precipitation (0.110) and snow water equivalent (0.103) have the least influence on model performance when their values are scrambling. Thus, they have little importance in predicting whether a lookout tower will be technically efficient or not. Appendix K provides a table of weights assigned to each model and a plot of AUROC scores against model weights. Furthermore, we use the five different criteria to classify a lookout as technically efficient or not technically efficient (first introduced in Chapter 5) to test the robustness of the feature importance values. A report of this robustness check is included in Appendix L.

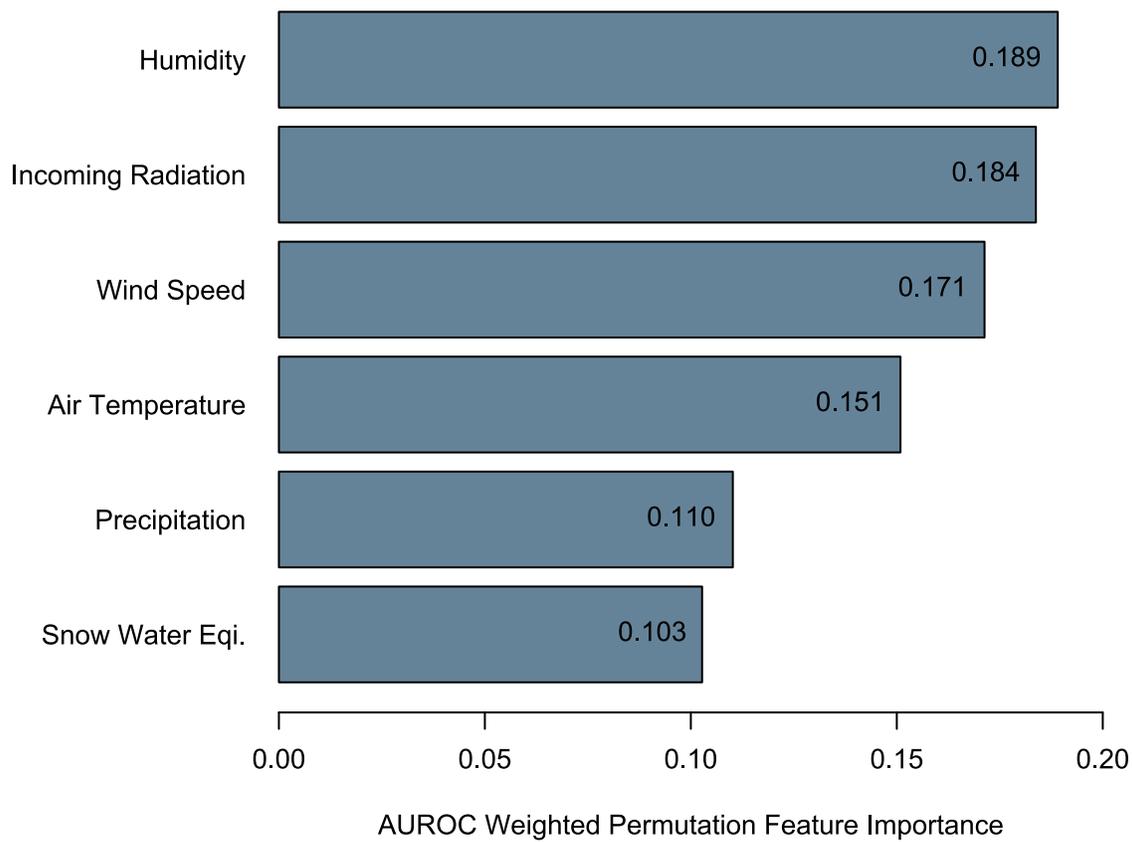


Figure 6.5 AUROC Weighted Permutation Feature Importance

Chapter 7. Discussion

Climate change and unpredictable atmospheric patterns will result in increasingly uncertain fire seasons. In challenging environments, there is a continuous need for resource optimization to make sure that wildfire policy is as efficient as possible. This research develops a production economics framework to examine the technical efficiency of wildfire detection by lookout towers. The framework considers the fixed visibility profile of lookouts as inputs and measures variations in outputs that capture the extensive and intensive margins of wildfire detection, i.e., changes in the proportion of fires detected by lookouts, and the speed of detection (reporting delays). The approach can be adapted to assess detection agents' performance in different settings. As such, our research develops a framework to assist wildfire management agencies make operational decisions and shape a superior detection system based on data-driven observations.

We estimate the technical efficiency of wildfire detection in Alberta. The Albertan detection system is composed mainly of traditional detection agents (e.g., air patrol, ground crew patrols, and lookout towers). Traditional detection agents have certain advantages that make them prevalent in many geographical regions. For instance, detection systems that incorporate lookout towers maximize the land area visible and lookout towers provide constant monitoring of wildlands (MNP LLP 2020). Public fire hotlines perform exceptionally well in areas that have higher population densities (Rego et al. 2013). Aerial patrols are quick in deployment, can cover large areas, access remote areas, and carry firefighting crews prepared to start fire suppression. Ground patrols crews provide rapid response to potential reports of wildfires.

On the other hand, traditional agents also have drawbacks. For instance, traditional technologies are manned modes of detection, therefore, they create room for human errors (Dampage et al. 2022; Yuan, Zhang and Liu 2015). When deployed, ground patrols are exposed to harmful air pollutants (Reisen, Hansen and Meyer 2011). Aerial reconnaissance for wildfire detection has high operational costs and is dangerous for personnel when visibility is low due to heavy wildfire smoke (Slavkovikj et al. 2014; Tzoumas et al. 2022).

Technological developments aligned with some of the negative aspects of traditional detection systems have led to the possibility of applying new technologies for wildfire detection. Remote sensing cameras, drones, deep-learning algorithms, and satellites are entering the market of monitoring wildlands and detecting wildfires (Bouguettaya et al. 2022; Allison et al. 2016; Zhang et al. 2019). However, these technologies come with their own set of challenges. For instance, the operation of drones is affected during conditions of heavy cloud cover or strong winds. Additionally, wildfire location can be accurately detected by drones only after the fire has reached a certain magnitude (Liu et al. 2022; Ichoku, Kahn and Chin 2012). The range of vision for machine-based lookout towers is lower than human-based watchtowers (Zhang et al. 2020). Therefore, a system of unmanned towers may require more lookout towers to be constructed to increase visibility coverage, working against the cost-effectiveness of replacing human observers with cameras (Zhang et al. 2020). Satellites can offer an alternative to cameras; however, they have the ability to survey the land ever so often. Satellites can be vulnerable to positional errors when detecting wildfires, leading to location estimates with errors of up to tens of kilometers (Liu et al. 2022). Current satellites that use infrared sensors have several gaps in observation, especially during afternoons and early evenings. Unfortunately, these long periods of blindness are during peak burn times when temperatures are high, winds are strong, and humidity levels favor extreme

wildfire behavior (Government of Canada and Canadian Space Agency 2022). All these factors need to be considered in light of a possible transition in detection technology.

Trade-offs for using different detection systems need to be closely examined to operate effectively. Lookout towers demand capital investments and, in a challenging fiscal environment, there is a need to consider their relevance (MNP LLP 2020). Lookout towers cost AB Wildfire approximately 7 million dollars (in total) during the five-year period of 2016 to 2020 (i.e., approximately 1.4 million annually). In 2019, the government budgeted \$6.9 million for the next five years to maintain and improve lookout towers. An alternative system (e.g. cameras) may be evaluated in terms of its technical efficiency, associated costs, and other factors in order to make benefit-cost comparisons.

Moreover, the thesis finds that weather patterns around the lookout towers during the fire season are important predictors of the technical efficiency classification, where the technical efficiency class denotes towers that are best-practice units and form the wildfire detection frontier. Recent research finds that new detection technologies are also influenced by weather. For instance, Zhang and co-authors find that weather influences the quality of camera and satellite images. Digital photos are good for detecting crown-fires while detecting ground or surface fires are still a challenge (Zhang et al. 2019)⁵. Even after minimizing these drawbacks, the incorporation of new technologies calls for testing phases which require trial and error and significant investments.

Furthermore, human-based lookout towers have been gaining attention given the expected changes in Alberta's labor and safety regulations. Working as an observer consists of working long

⁵ Crown-fires burn the forest canopy which is made of foliage, branches, and fuels above the surface level. Surface fires burn fuels on the surface such as litter, duff, and forest residuals. Ground fires occur underground or below the surface and burn peat, dead vegetation, and move very slowly but are harder to suppress (These definitions are referred to from AB Wildfire's detection data dictionary).

hours in isolation that can influence the performance of observers due to fatigue. With new labor regulations, decision-makers may have to hire more than one observer per tower, doubling the cost of manpower at the very least and further affecting the cost-effectiveness of lookout towers (MNP LLP 2020). More stringent hiring practices in the future can make operations difficult in an increasingly challenging surveillance environment. As institutional settings change, there is a need for continuous studies of the performance of lookout towers and detection systems.

But institutions are not the only varying factor, and the future of wildfire detection must consider the changing global climate. The process of detecting wildfires has gained significant attention due to the growing concerns about frequency and severity of wildfires (Johnston et al. 2020). Previous studies highlight one common point: the importance of early detection in mitigating the damages cause by wildfires (Diendéré and Kaboré 2023; McFayden et al. 2019; Lindenmayer et al. 2022). But what drives technical efficiency? The latter part of this study sheds light on the influence of weather variables in predicting technical efficiency of wildfire detection. Further understanding the influence of meteorological variables on technical efficiency can assist in making informed decisions about resource allocation. Understanding weather and how it affects lookout towers may guide decision-making about the right time for lookout tower maintenance, operational times, and which towers are to be manned more than others.

This work also raises questions for future studies. For example, more research is needed to bridge the gap between wildfire science and production economics. Such research would facilitate the assessment of how new wildfire detection technologies compare to the existing lookout systems. In that, bias corrected technical efficiency analysis of data from other jurisdictions (e.g., those using drones, cameras, and/or other alternative technologies discussed above) is an important step to inform costs and benefits of varying detection system configurations. The envelopment

techniques employed in this study offer an opportunity to assess how different detection agents may perform as part of a mixed-technology detection system. With more detailed data, cameras, for example, can be directly compared to manned lookouts in a technical efficiency framework.

7.1. Limitations

This work has several limitations. First, DEA estimates of technical efficiency are sensitive to measurement errors and outliers. Nonparametric models used to estimate technical efficiency do not require assumptions about the shape of the production technology. Envelopment estimators like DEA rely on the enveloping data to estimate the production frontier. While this nonparametric strategy avoids production function misspecification errors, the nonparametric envelopment makes the DEA frontier sensitive to measurement errors and outliers. For instance, while referring to lightning maps is a good practice for estimating fire start times, these are often estimated by crews on the ground. Therefore, reporting delay (the difference between fire start time and fire reporting time) may be measured with error. However, as long as these measurement errors are random in the sense that they are not associated with the visibility profile of towers, our technical efficiency measures are still informative of the performance of the towers. A short discussion of outliers and how they can affect envelopment estimates is included in Appendix F.

Second, there are limitations related to the weather data. The data is used to build machine learning predictors for lookout tower technical efficiency. As depicted in Figure H.1, numerous weather stations collect weather data, signifying that local weather is captured accurately. The stations capture the main meteorological variables that are relevant for wildfire behavior, however,

many observations are missing. To capture local weather around lookout towers and reduce missing observations, we applied an inverse distance weighted approach as discussed in the previous chapter. Additionally, only a few weather stations have the tools and sensors that provide the ability to detect incoming radiation. For stations without these sensors, the incoming radiation observations are modelled using extraterrestrial radiation, temperature, and station longitude and latitude. A summary of how ACIS models incoming radiation values is presented in Appendix I. While the estimations show a high agreement with real observed data, there is room for improvement by incorporating higher resolution of data and non-estimated data.

Third, the inputs and outputs used in this research represent one way of characterizing lookout towers as production units. Lookout towers use their visibility profile to produce the maximum number of timely wildfire detections. The inputs, areas that are directly and/or indirectly visible from a lookout tower's viewshed analysis, are fixed and not changing. Thus, inputs used in our study are not time-varying inputs. Given that the inputs we consider are fixed, our realization of the production function of lookout towers can be classified as short-run production. Different perspectives for the characterization of wildfire detection production functions are possible. Models of time-varying inputs can consider, for instance, the experience of lookout tower observers, fatigue and other characteristics of the observer, and staffing decisions of lookout towers indicating when a given lookout tower is staffed and active. A detection model with time varying inputs gives us a long-run perspective of the wildfire detection production function. Appendix M includes a discussion on the difference between short-run and long-run production functions. Moreover, it is also important to acknowledge that misspecification of inputs can impact the DEA estimate of technical efficiency. Our focus is on output-oriented technical efficiency given

visibility profiles. As a result, models with richer datasets that include, for example, fire and observer characteristics may produce different DEA estimates (Smith 1997).

Fourth, we select AUROC as the metric to compare different models' performance and construct the AUROC Weighted Permutation Feature Importance values. While AUROC is a widely used metric of performance and it indicates how well the ML model can differentiate between class labels, it tends to have drawbacks depending on the characteristics of the dataset (Mungo et al. 2023). Specifically, the AUROC is problematic when a dataset does not have equal distribution of classes, i.e., when it is considered to be an unbalanced dataset. To see this, note if a binary classification dataset has high density for a particular class (i.e., low density for the other class), the ML model can easily achieve high accuracy by predicting the prevalent class correctly. In these cases, because there is only a small number of observations in the low-density class (by definition), the ML model may perform poorly in predicting/classifying the data. While the actual dataset for our analysis is not heavily unbalanced with approximately 60% of technically efficient towers (40% not technically efficient), the sensitive characteristic of the AUROC metric should be kept in mind. The issue may not be prevalent in our analysis, but it is important to highlight the limitation that our dataset is not fully balanced.

Bibliography

- AB Wildfire. 2022. "Lookout Observer Manual."
- AB Wildfire. 2020. "Wildfire Management Data Dictionary."
- AB Wildfire, and Government of Alberta. 2022. "Whitecourt Area Update." Available at: <https://srd.web.alberta.ca/whitecourt-area-update/may-2022-05-26> [Accessed October 14, 2023].
- Abatzoglou, J.T., A.P. Williams, and R. Barbero. 2019. "Global Emergence of Anthropogenic Climate Change in Fire Weather Indices." *Geophysical Research Letters* 46(1):326–336.
- ACIS. 2019. "Alberta Climate Information Service (ACIS) Definitions Disclaimer."
- Adamecz-Völgyi, A., M. Henderson, and N. Shure. 2020. "Is 'first in family' a good indicator for widening university participation?" *Economics of Education Review* 78.
- Aigner, D., C.A.K. Lovell, and P. Schmidt. 1977. "FORMULATION AND ESTIMATION OF STOCHASTIC FRONTIER PRODUCTION FUNCTION MODELS*." *Journal of Econometrics*.
- Alberta Climate Information Service. 2020. "Current and Historical Alberta Weather Station Data Viewer." Available at: <https://acis.alberta.ca/acis/weather-data-viewer.jsp> [Accessed October 2, 2023].
- Alberta Climate Information Service. 2019. "Methodology and Data Sources for Agriculture and Forestry's Interpolated Data."
- Alberta Wildfire, and FireSmart. 2020. *Farm and Acreage: A Guide to Reducing the Risk From Wildfire*. Alberta Wildfire. Available at: <https://wildfire.alberta.ca/firesmart/firesmart-communities/documents/farm-and-ranch-magazine-aug-25.pdf> [Accessed January 3, 2023].
- Aldamak, A., and S. Zolfaghari. 2017. "Review of efficiency ranking methods in data envelopment analysis." *Measurement: Journal of the International Measurement Confederation* 106:161–172.
- Alfaro, E., M. Gámez, and N. García. 2013. "adabag: An R Package for Classification with Boosting and Bagging." *Journal of Statistical Software* 54(2):1–35.
- Allison, R.S., J.M. Johnston, G. Craig, and S. Jennings. 2016. "Airborne optical and thermal remote sensing for wildfire detection and monitoring." *Sensors (Switzerland)* 16(8).

- Alpaydin, E. 2020. *Introduction to Machine Learning* 4th ed. F. Bach, ed. MIT Press.
- Amiri, T., A. Banj Shafiei, M. Erfanian, O. Hosseinzadeh, and H. Beygi Heidarlou. 2022. "Using forest fire experts' opinions and GIS/remote sensing techniques in locating forest fire lookout towers." *Applied Geomatics*.
- Badunenko, O., and P. Mozharovskyi. 2016. "Nonparametric frontier analysis using Stata." *Stata Journal* 16(3):550–589.
- Bandunenko, O., P. Mozharovskyi, and Y. Kolomiytseva. 2022. "nspf: Nonparametric and Stochastic Efficiency and Productivity." Available at: <https://cran.r-project.org/web/packages/nspf/nspf.pdf> [Accessed October 25, 2023].
- Banerjee, T., W. Heilman, S. Goodrick, J.K. Hiers, and R. Linn. 2020. "Effects of canopy midstory management and fuel moisture on wildfire behavior." *Scientific Reports* 10(1).
- Bao, S., N. Xiao, Z. Lai, H. Zhang, and C. Kim. 2015. "Optimizing watchtower locations for forest fire monitoring using location models." *Fire Safety Journal* 71:100–109.
- Barmpoutis, P., K. Dimitropoulos, K. Kaza, and N. Grammalidis. 2019. "Fire Detection from Images Using Faster R-CNN and Multidimensional Texture Analysis." *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Bergstra, J., J.B. Ca, and Y.B. Ca. 2012. "Random Search for Hyper-Parameter Optimization Yoshua Bengio." Available at: <http://scikit-learn.sourceforge.net>.
- Bernath, P., C. Boone, and J. Crouse. 2022. "Wildfire smoke destroys stratospheric ozone." *Science* 375(6586):1292–1295. Available at: www.ipcc.ch/report/ar6/wg1/.
- Berrar, D. 2018. "Cross-validation." In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*. Elsevier, pp. 542–545.
- Bishop, M.C. 2006. *Pattern Recognition and Machine Learning* M. Jordan, J. Kleinberg, and B. Scholkopf, eds.
- Bouguettaya, A., H. Zarzour, A.M. Taberkit, and A. Kechida. 2022. "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms." *Signal Processing* 190.
- Breiman, L. 2001. "Random Forests." *Machine Learning* 45:5–32.
- Canepa, A., and F. Drogo. 2021. "Wildfire crime, apprehension and social vulnerability in Italy." *Forest Policy and Economics* 122.
- Castro, N., M.A. Akhloufi, and A. Couturier. 2018. "UAVs for wildland fires." In *SPIE-Intl Soc Optical Eng*, p. 23.

- Chambers, R.G. 1988. *Applied Production Analysis - A Dual Approach*.
- Charnes, A., W. Cooper, and E. Rhodes. 1978. "Measuring the efficiency of decision making units."
- Chen, T., T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, Y. Li, and J. Yuan. 2023. "xgboost: Extreme Gradient Boosting." Available at: <https://CRAN.R-project.org/package=xgboost>.
- CNN Business. 2023. "Moody's estimates Hawaiian wildfires caused up to \$6 billion in economic losses." Available online at <https://www.cnn.com/2023/08/22/business/maui-wildfires-moodys-economic-damages/index.html#:~:text=Moody's%20RMS%20estimated%20up%20to,%246%20billion%2C%20Moody's%20said%20Tuesday>. Accessed on Nov 29, 2023.
- Çoban, H.O., and H. Bereket. 2020. "Visibility analysis of fire lookout towers protecting the mediterranean forest ecosystems in Turkey." *Sumarski List* 144(7–8):393–407.
- Coelli, T.J., D.S.P. Rao, C.J. O'Donnell, and G.E. Battese. 2005. *An introduction to efficiency and productivity analysis* Second. Springer.
- Collins, L., P. Griffioen, G. Newell, and A. Mellor. 2018. "The utility of Random Forests for wildfire severity mapping." *Remote Sensing of Environment* 216:374–384.
- Cooper W. William, L.M. Seiford, and J. Zhu. 2004. *HANDBOOK ON DATA ENVELOPMENT ANALYSIS* R. Allen, T. R. Anderson, A. D. Athanassopoulos, R. D. Banker, D. Barr, J. A. Chilingerian, W. Cook, H. Deng, R. Fare, S. Grosskopf, Z. Huang, S. Li, S. X. Li, R. Natarajan, J. C. Paradi, M. Portela, J. Ruggiero, D. Sherman, L. Simar, E. Thanassoulis, K. Tone, K. Triantis, S. Vela, G. Whittake, and Z. Yang, eds. Kluwer's Academic Publishers.
- Cruz, M.G., and M.E. Alexander. 2019. "The 10% wind speed rule of thumb for estimating a wildfire's forward rate of spread in forests and shrublands." *Annals of Forest Science* 76(2).
- Curtis, P.G., C.M. Slay, N.L. Harris, A. Tyukavina, and M.C. Hansen. 2018. "Classifying drivers of global forest loss." *Science* 361(6407). Available at: <https://www.science.org>.
- Dampage, U., L. Bandaranayake, R. Wanasinghe, K. Kottahachchi, and B. Jayasanka. 2022. "Forest fire detection system using wireless sensor networks and machine learning." *Scientific Reports* 12(1).
- Dhungana, B.R., P.L. Nuthall, and G. v Nartea. 2004. "Measuring the economic inefficiency of Nepalese rice farms using data envelopment analysis." *The Australian Journal of Agricultural and Resource Economics* 48(2):347–369.

- Diendéré, A.A., and D. Kaboré. 2023. “Preferences for a payment for ecosystem services program to control forest fires in Burkina Faso: A choice experiment.” *Forest Policy and Economics* 151.
- Dong, H., H. Wu, P. Sun, and Y. Ding. 2022. “Wildfire Prediction Model Based on Spatial and Temporal Characteristics: A Case Study of a Wildfire in Portugal’s Montesinho Natural Park.” *Sustainability (Switzerland)* 14(16).
- Dutta, R., A. Das, and J. Aryal. 2016. “Big data integration shows Australian bush-fire frequency is increasing significantly.” *Royal Society Open Science* 3(2).
- Eltahir, E.A.B. 1998. “A soil moisture-rainfall feedback mechanism.” *Water Resources Research* 34(4):765–776.
- English, W. 2018. “How the 30-30-30 Crossover Rule affects the threat of a wildfire sparking.” *KelownaNow*. Available at: https://www.kelownanow.com/watercooler/news/news/Okanagan/How_the_30_30_30_Crossover_Rule_affects_the_threat_of_a_wildfire_sparking/ [Accessed August 27, 2023].
- Fang, K., Q. Yao, Z. Guo, B. Zheng, J. Du, F. Qi, P. Yan, J. Li, T. Ou, J. Liu, M. He, and V. Trouet. 2021. “ENSO modulates wildfire activity in China.” *Nature Communications* 12(1).
- Fantini, S., M. Fois, R. Secci, P. Casula, G. Fenu, and G. Bacchetta. 2022. “Incorporating the visibility analysis of fire lookouts for old-growth wood fire risk reduction in the Mediterranean island of Sardinia.” *Geocarto International*.
- Farrell, M.J. 1957. “The Measurement of Productive Efficiency.” *Source: Journal of the Royal Statistical Society. Series A (General)* 120(3):253–290.
- Fernandes, P.M., A.M.G. Barros, A. Pinto, and J.A. Santos. 2016. “Characteristics and controls of extremely large wildfires in the western Mediterranean Basin.” *Journal of Geophysical Research: Biogeosciences* 121(8):2141–2157.
- Flannigan, M.D., M.A. Krawchuk, W.J. De Groot, B.M. Wotton, and L.M. Gowman. 2009. “Implications of changing climate for global wildland fire.” *International Journal of Wildland Fire* 18(5):483–507.
- Friedman, J.H. 2001. “Greedy Function Approximation: A Gradient Boosting Machine.”
- Géron, A. 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems* 2nd ed.
- Gibson, R., T. Danaher, W. Hehir, and L. Collins. 2020. “A remote sensing approach to mapping fire severity in south-eastern Australia using sentinel 2 and random forest.” *Remote Sensing of Environment* 240.

- Goodfellow, I., Y. Bengio, and A. Courville. 2017. *Deep Learning*. MIT Press.
- Google Developers. 2023. “Machine Learning Foundational Courses.” Available at: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc> [Accessed October 2, 2023].
- Government of Canada, and Canadian Space Agency. 2022. “WildFireSat: Enhancing Canada’s ability to manage wildfires.” Available at: <https://www.asc-csa.gc.ca/eng/satellites/wildfiresat/> [Accessed January 1, 2023].
- Gutiérrez, E., and S. Lozano. 2013. “Avoidable damage assessment of forest fires in European countries: An efficient frontier approach.” *European Journal of Forest Research* 132(1):9–21.
- Halofsky, J.E., D.L. Peterson, and B.J. Harvey. 2020. “Changing wildfire, changing forests: the effects of climate change on fire regimes and vegetation in the Pacific Northwest, USA.” *Fire Ecology* 16(1).
- Hanes, C.C., X. Wang, P. Jain, M.A. Parisien, J.M. Little, and M.D. Flannigan. 2019. “Fire-regime changes in Canada over the last half century.” *Canadian Journal of Forest Research* 49(3):256–269.
- Hastie, T., R. Tibshirani, and J. Friedman. 2001. *Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction* 2ND ed.
- Herawati, H., and H. Santoso. 2011. “Tropical forest susceptibility to and risk of fire under changing climate: A review of fire nature, policy and institutions in Indonesia.” *Forest Policy and Economics* 13(4):227–233.
- Hjalmarsson, L., S.C. Kumbhakar, and A. Heshmati. 1996. “DEA, DFA and SFA: A Comparison.” Kluwer Academic Publishers.
- Homburg, C. 2001. “Using data envelopment analysis to benchmark activities.”
- Ichoku, C., R. Kahn, and M. Chin. 2012. “Satellite contributions to the quantitative characterization of biomass burning for climate modeling.” *Atmospheric Research* 111:1–28.
- Jain, P., D. Castellanos-Acuna, S.C.P. Coogan, J.T. Abatzoglou, and M.D. Flannigan. 2022. “Observed increases in extreme fire weather driven by atmospheric humidity and temperature.” *Nature Climate Change* 12(1):63–70.
- James, G. (Gareth M.), D. Witten, T. Hastie, and R. Tibshirani. 2013. *An introduction to statistical learning : with applications in R* 7th ed.

- Janabi, S., I. Shourbaji, and M. Salman. 2018. "Assessing the suitability of soft computing approaches for forest fires prediction." *Applied Computing and Informatics* 14(2):214–224.
- Johnston, L.M., X. Wang, S. Erni, S.W. Taylor, C.B. McFayden, J.A. Oliver, C. Stockdale, A. Christianson, Y. Boulanger, S. Gauthier, D. Arseneault, B.M. Wotton, M.A. Parisien, and M.D. Flannigan. 2020. "Wildland fire risk research in Canada." *Environmental Reviews* 28(2):164–186.
- Karatzoglou, A., A. Smola, and K. Hornik. 2023. "kernlab: Kernel-Based Machine Learning Lab." Available at: <https://CRAN.R-project.org/package=kernlab>.
- Karatzoglou, A., A. Smola, K. Hornik, and A. Zeileis. 2004. "kernlab – An S4 Package for Kernel Methods in R." *Journal of Statistical Software* 11(9):1–20.
- Korostelev, A.P., L. Simar, and A.B. Tsybakov. 1995a. "Efficient Estimation of Monotone Boundaries."
- Korostelev, A.P., L. Simar, and A.B. Tsybakov. 1995b. "On Estimation of Monotone and Convex Boundaries." *Pub. Inst. Stat. Univ. Paris XXXIX, fasc. 1*:3–18.
- Krueger, E.S., T.E. Ochsner, D.M. Engle, J.D. Carlson, D. Twidwell, and S.D. Fuhlendorf. 2015. "Soil Moisture Affects Growing-Season Wildfire Size in the Southern Great Plains." *Soil Science Society of America Journal* 79(6):1567–1576.
- Kucuk, O., O. Topaloglu, A.O. Altunel, and M. Cetin. 2017. "Visibility analysis of fire lookout towers in the Boyabat State Forest Enterprise in Turkey." *Environmental Monitoring and Assessment* 189(7).
- Kuhn, M., J. Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, R Core Team, M. Benesty, R. Lescarbeau, A. Ziem, L. Scrucca, Y. Tang, C. Candan, and T. Hunt. 2023. "caret: Classification and Regression Training." Available at: <https://cran.r-project.org/web/packages/caret/caret.pdf> [Accessed October 25, 2023].
- Kuhn, and Max. 2008. "Building Predictive Models in R Using the caret Package." *Journal of Statistical Software* 28(5):1–26. Available at: <https://www.jstatsoft.org/index.php/jss/article/view/v028i05>.
- Lawson, B.D., and O.B. Armitage. 2008. "Weather Guide for the Canadian Forest Fire Danger Rating System." Available at: <https://scf.rncan.gc.ca/pubwarehouse/pdfs/29152.pdf> [Accessed August 8, 2023].
- Lindenmayer, D., P. Zylstra, and M. Yebra. 2022. "Adaptive wildfire mitigation approaches." *Science* 377(6611):1163.

- Liu, Q., X. Li, H. Zhang, J. Ren, S. Yang, L. Cao, J. Liang, and S. Ling. 2022. “IntelliSense silk fibroin ionotronic batteries for wildfire detection and alarm.” *Nano Energy* 101.
- Long, L.K., L. van Thap, N.T. Hoai, and T.T.T. Pham. 2020. “Data envelopment analysis for analyzing technical efficiency in aquaculture: The bootstrap methods.” *Aquaculture Economics and Management* 24(4):422–446.
- Lookout Observer Manual. 2022. “Lookout Observer Manual.”
- Mahdavi, D. 2023. “Millions of Canadians will face extreme fire danger this summer. Here’s what that means and how to stay safe.” *CBC*. Available at: <https://www.cbc.ca/news/canada/thunder-bay/interpreting-fire-danger-scale-1.6818729#:~:text=A%20good%20guide%20for%20fire,be%20extra%20cautious%20about%20ignition.> [Accessed August 27, 2023].
- Malmquist, S. 1953. “Index numbers and indifference surfaces.” *Trabajos de Estadística* 4.
- McFayden, C.B., D.G. Woolford, A. Stacey, D. Boychuk, J.M. Johnston, M.J. Wheatley, and D.L. Martell. 2019. “Risk assessment for wildland fire aerial detection patrol route planning in Ontario, Canada.” *International Journal of Wildland Fire* 29(1):28–41.
- McGrath, D., L. Zeller, R. Bonnell, W. Reis, S. Kampf, K. Williams, M. Okal, A. Olsen-Mikitowicz, E. Bump, M. Sears, and K. Rittger. 2023. “Declines in Peak Snow Water Equivalent and Elevated Snowmelt Rates Following the 2020 Cameron Peak Wildfire in Northern Colorado.” *Geophysical Research Letters* 50(6).
- Meeusen, W., J. Van, and D. Broeck. 1977. “Efficiency Estimation from Cobb-Douglas Production Functions with Composed Error.” Available at: <https://about.jstor.org/terms>.
- Mendes, I. 2010a. “A theoretical economic model for choosing efficient wildfire suppression strategies.” *Forest Policy and Economics* 12(5):323–329.
- Mendes, I. 2010b. “A theoretical economic model for choosing efficient wildfire suppression strategies.” *Forest Policy and Economics* 12(5):323–329.
- MNP LLP. 2020. “Spring 2019 Wildfire Review Final Report.”
- Molnar, C. 2020. “Interpretable Machine Learning A Guide for Making Black Box Models Explainable.” Available at: [www.dbooks.orghttp://leanpub.com/interpretable-machine-learning](http://leanpub.com/interpretable-machine-learning).
- Moreira, F., and G. Pe’er. 2018. “Agricultural policy can reduce wildfires.” *Science* 359(6379):1001.

- Mungo, L., F. Lafond, P. Astudillo-Estévez, and J.D. Farmer. 2023. "Reconstructing production networks using machine learning." *Journal of Economic Dynamics and Control* 148.
- O, S., X. Hou, and R. Orth. 2020. "Observational evidence of wildfire-promoting soil moisture anomalies." *Scientific Reports* 10(1).
- Pedregosa, F., V. Michel, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, D. Cournapeau, F. Pedregosa, G. Varoquaux, A. Gramfort, B. Thirion, O. Grisel, V. Dubourg, A. Passos, M. Brucher, M. Perrot, and É. Duchesnay. 2011. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12:2825–2830. Available at: <http://scikit-learn.sourceforge.net>.
- Pompa-García, M., R. Solís-Moreno, E. Rodríguez-Téllez, A. Pinedo-Álvarez, D. Avila-Flores, C. Hernández-Díaz, and E. Velasco-Bautista. 2010. "Viewshed Analysis for Improving the Effectiveness of Watchtowers, in the North of Mexico."
- Rego, F., F.X. Catry, C. Montiel, and O. Karlsson. 2013. "Influence of territorial variables on the performance of wildfire detection systems in the Iberian Peninsula." *Forest Policy and Economics* 29:26–35.
- Rego, F.C., and F.X. Catry. 2006. "Modelling the effects of distance on the probability of fire detection from lookouts." *International Journal of Wildland Fire* 15(2):197–202.
- Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat. 2019. "Deep learning and process understanding for data-driven Earth system science." *Nature* 566(7743):195–204.
- Reisen, F., D. Hansen, and (Mick) P. Meyer. 2011. "Exposure to bushfire smoke during prescribed burns and wildfires: Firefighters' exposure risks and options." *Environment International* 37(2):314–321.
- Rodríguez, J.D., A. Pérez, and J.A. Lozano. 2010. "Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(3):569–575.
- Sakellariou, S., G. Sfoungaris, and O. Christopoulou. 2022. "Territorial Resilience Through Visibility Analysis for Immediate Detection of Wildfires Integrating Fire Susceptibility, Geographical Features, and Optimization Methods." *International Journal of Disaster Risk Science* 13(4):621–635.
- Sakr, G.E., I.H. Elhajj, G. Mitri, and U.C. Wejinya. 2010. "Artificial Intelligence for Forest Fire Prediction." In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*.

- Schliep, K., and K. Hechenbichler. 2016. “kkn: Weighted k-Nearest Neighbors.” Available at: <https://CRAN.R-project.org/package=kkn>.
- Schmidt, P. 1985. “Frontier production functions.” *Econometric Reviews* 4(2):289–328.
- Schmidt, P., and C.E. Lemke. 1976. “On the Statistical Estimation of Parametric Frontier Production Functions.”
- Scholten, R.C., R. Jandt, E.A. Miller, B.M. Rogers, and S. Veraverbeke. 2021. “Overwintering fires in boreal forests.” *Nature* 593(7859):399–404.
- Sharma, S., J.D. Carlson, E.S. Krueger, D.M. Engle, D. Twidwell, S.D. Fuhlendorf, A. Patrignani, L. Feng, and T.E. Ochsner. 2020. “Soil moisture as an indicator of growing-season herbaceous fuel moisture and curing rate in grasslands.” *International Journal of Wildland Fire* 30(1):57–69.
- Shephard, R.W. 1981. *Cost and Production Functions*. Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: <http://link.springer.com/10.1007/978-3-642-51578-1>.
- Simar, L., and P.W. Wilson. 2000. “A general methodology for bootstrapping in non-parametric frontier models.” *Journal of Applied Statistics* 27(6):779–802.
- Simar, L., and P.W. Wilson. 2007. “Estimation and inference in two-stage, semi-parametric models of production processes.” *Journal of Econometrics* 136(1):31–64.
- Simar, L., and P.W. Wilson. 1998. “Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models.” *Management Science* 44(1):49–61.
- Slavkovikj, V., S. Verstockt, S. van Hoecke, and R. van de Walle. 2014. “Review of wildfire detection using social media.” *Fire Safety Journal* 68:109–118.
- Smith, P. “Model misspecification in Data Envelopment Analysis.”
- Sousa, M. da C.S. de, and B. Stosic. 2005. “Technical Efficiency of the Brazilian Municipalities: Correcting Nonparametric Frontier Measurements for Outliers.”
- Steele, T.W., and J.C. Stier. 1998. “An Economic Evaluation of Public and Organized Wildfire Detection in Wisconsin.”
- Stojanova, D., A. Kobler, P. Ogrinc, B. Ženko, and S. Džeroski. 2012. “Estimating the risk of fire outbreaks in the natural environment.” *Data Mining and Knowledge Discovery* 24(2):411–442.
- Stosic, B., and M. Sousa. 2003. “Jackstrapping DEA Scores for Robust Efficiency Measurement.”

- Suleiman, A., M.R. Tight, and A.D. Quinn. 2016. “Hybrid Neural Networks and Boosted Regression Tree Models for Predicting Roadside Particulate Matter.” *Environmental Modeling and Assessment* 21(6):731–750.
- Therneau, T., and B. Atkinson. 2022. “rpart: Recursive Partitioning and Regression Trees.” Available at: <https://CRAN.R-project.org/package=rpart>.
- Tuszynski, J. 2021. “caTools: Tools: Moving Window Statistics, GIF, Base64, ROC AUC, etc.” Available at: <https://CRAN.R-project.org/package=caTools>.
- Tzoumas, G., L. Pitonakova, L. Salinas, C. Scales, T. Richardson, and S. Hauert. 2022. “Wildfire detection in large-scale environments using force-based control for swarms of UAVs.” *Swarm Intelligence*.
- Venables, W.N., and B.D. Ripley. 2002. *Modern Applied Statistics with S* Fourth. New York: Springer. Available at: <https://www.stats.ox.ac.uk/pub/MASS4/>.
- Wagner, V.C.E. 1987. “Development and Structure of the Canadian Forest Fire Weather Index System.” *Canadian Forestry Service Headquarters*. Available at: <https://d1ied5g1xfqpx8.cloudfront.net/pdfs/19927.pdf> [Accessed August 3, 2023].
- Walker, X.J., J.L. Baltzer, S.G. Cumming, N.J. Day, C. Ebert, S. Goetz, J.F. Johnstone, S. Potter, B.M. Rogers, E.A.G. Schuur, M.R. Turetsky, and M.C. Mack. 2019. “Increasing wildfires threaten historic carbon sink of boreal forest soils.” *Nature* 572(7770):520–523.
- Weihs, C., U. Ligges, K. Luebke, and N. Raabe. 2005. “klaR Analyzing German Business Cycles.” In D. Baier, R. Decker, and L. Schmidt-Thieme, eds. *Data Analysis and Decision Support*. Berlin: Springer-Verlag, pp. 335–343.
- Westerling, A.L., H.G. Hidalgo, D.R. Cayan, and T.W. Swetnam. 2006. “Warming and Earlier Spring Increase Western U.S. Forest Wildfire Activity.” *Science* 313(5789):936–940.
- Wright, M.N., and A. Ziegler. 2017. “ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R.” *Journal of Statistical Software* 77(1):1–17.
- Wu, J., X.Y. Chen, H. Zhang, L.D. Xiong, H. Lei, and S.H. Deng. 2019. “Hyperparameter optimization for machine learning models based on Bayesian optimization.” *Journal of Electronic Science and Technology* 17(1):26–40.
- Xie, L., R. Zhang, J. Zhan, S. Li, A. Shama, R. Zhan, T. Wang, J. Lv, X. Bao, and R. Wu. 2022. “Wildfire Risk Assessment in Liangshan Prefecture, China Based on An Integration Machine Learning Algorithm.” *Remote Sensing* 14(18).
- Yu, P., O.B. Toon, C.G. Bardeen, Y. Zhu, K.H. Rosenlof, R.W. Portmann, T.D. Thornberry, R.-S. Gao, S.M. Davis, E.T. Wolf, J. de Gouw, D.A. Peterson, M.D. Fromm, and A. Robock.

2019. "Black carbon lofts wildfire smoke high into the stratosphere to form a persistent plume." Available at: <https://www.science.org>.
- Yuan, C., Y. Zhang, and Z. Liu. 2015. "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques." *Canadian Journal of Forest Research* 45(7):783–792.
- Zhang, F., P. Zhao, J. Thiyagalingam, and T. Kirubarajan. 2019. "Terrain-influenced incremental watchtower expansion for wildfire detection." *Science of the Total Environment* 654:164–176.
- Zhang, F., P. Zhao, S. Xu, Y. Wu, X. Yang, and Y. Zhang. 2020. "Integrating multiple factors to optimize watchtower deployment for wildfire detection." *Science of the Total Environment* 737.
- Zhang, Q.X., G.H. Lin, Y.M. Zhang, G. Xu, and J.J. Wang. 2018. "Wildland Forest Fire Smoke Detection Based on Faster R-CNN using Synthetic Smoke Images." In *Procedia Engineering*. Elsevier Ltd, pp. 441–446.
- Zhao, Y., J. Ma, X. Li, and J. Zhang. 2018. "Saliency detection and deep learning-based wildfire identification in uav imagery." *Sensors (Switzerland)* 18(3).

Appendix A: Forest Protection Area in Alberta

Forest Protection Area (FPA) are geographical areas assigned by the government. The aim is to protect these lands by mitigating damage from wildfires, invasive insects, and other negative factors. Figure A.1 depicts the FPA for the province of Alberta. The pink area is the designated FPA in which AB Wildfire actively detects, monitors, and suppresses wildfires.



Figure A.1 Forest Protection Areas of Alberta (Source: AB Wildfire and Government of Alberta (2022))

Appendix B: Fires outside Surveillance Regions

Figure B.1 plots the yearly frequency of wildfire by their class sizes. The magnitude of the entire bar represents the frequency of all wildfires that originated outside the surveillance region of the lookout towers. While the number of fires outside the surveillance region shows an increasing trend, most of these reported fires are of the size class of A, the smallest size (no bigger than 0.10 ha). As such, it can be said that detection agents are identifying signs of potential fires early as compared to fires going undetected for longer till they grow into larger wildfires.



Figure B.1 Frequency of Wildfires outside Surveillance Regions, by Size Class and Year

Appendix C: Output-oriented DEA Model (Variable Returns to Scale)

The estimates of DEA technical efficiency results presented in Chapter 5 are Debreu-Farrell measures of technical efficiency. We use the model formulations as discussed by Coelli et al. (2005). This study uses the output-oriented model with variable returns to scale. In the output-oriented method, the program aims to identify technical inefficiency as the proportion expansion needed in detection output production with input levels being constant. The decision to pick an output-oriented model is clear as all lookout towers utilize their own fixed levels of visibility areas (does not change over time) as inputs in the production of detection outputs.

Lookout towers as data points are represented by i , ($i = 1, \dots, I$). N inputs are denoted by vector $x_i = (x_{i1}, \dots, x_{iN})$ where $N \in \mathbb{R}^N$ while vector $y_i = (y_{i1}, \dots, y_{iM})$ for $M \in \mathbb{R}^M$ denotes the outputs. Thus, the input matrix X and output matrix Y ($M \times I$) represent the data for I observations. θ is a scalar and z ($I \times 1$) is a vector of constants. Therefore, for I data points, M outputs, and N inputs, the measure of output-oriented technical efficiency is calculated as:

$$\begin{aligned} & \max_{\theta, z} \theta \text{ subject to} \\ & -y_{im}\theta_m + \sum_{i=1}^I z_i y_{im} \geq 0, \\ & x_{in} - \sum_{i=1}^I z_i x_{in} \geq 0, \\ & \sum_{i=1}^I z_i = 1, \\ & z_i \geq 0, \\ & m = 1, \dots, M, \\ & n = 1, \dots, N \end{aligned}$$

Where $1 \leq \theta \leq \infty$, in which $\theta - 1$ is the radial increase in detection output while holding inputs fixed. For example, if $\theta = 1.08$, the lookout tower needs to improve detection by 8% (i.e., $1.08 - 1 = 0.08$) to reach the technical efficiency frontier.

Appendix D: Types of Bootstrapping

For estimation purposes of this thesis, we used the *teradialbc* command from the *nspf* R package (Bandunenko, Mozharovskyi and Kolomyitseva 2022). There are many types of bootstrapping techniques, and the bootstrapping procedure can follow different rules for picking sub-samples from the given dataset. In this study we consider two types of bootstraps: (1) homogenous or (2) heterogenous.

As explained by Badunenko and Mozharovskyi, the two approaches differ in the way pseudo sub-samples are constructed. Homogenous bootstrap is straightforward and resamples from the original dataset with replacement. The average of the statistic of interest (for instance technical efficiency scores) is computed which is then compared to the original statistic, thus explaining the uncertainty of the original estimates. The heterogenous bootstrap divides the given data into many subgroups based on a criterion. These sub-groups are resampled to make a new sample. The process is repeated many times till a distribution of the estimates is formed. Both bootstrap approaches can be “smoothed”. A smoothed bootstrap simply forms a smooth version of the original dataset by considering all small fluctuations in the data (Badunenko and Mozharovskyi 2016). We use the smoothed homogenous bootstrap to calculate the bias-corrected technical efficiency estimates for the lookout towers.

Appendix E: Yearly Distribution of Technical Efficiency Scores

Figure E.1 provides histograms of the yearly distribution of bias-corrected technical efficiency estimates for model A. It can be noted that all the histograms are positively skewed (right skewed). Most of the technical efficiency estimates lie closer to one. Therefore, it can be said that most of the lookout towers in a year are very close to the technical efficiency frontier. To clearly visualize the behaviour of the bias-corrected measures of technical efficiency estimates, we chose 1.20 as the cut-off value for x-axis. Any lookout tower that can increase its production of detection outputs, holding the visibility areas in their SR fixed, by more than 20% is not included in Figure E.1. These lookout towers are considered in all other analysis and make up 5% of the total number of observations at the tower-year level.

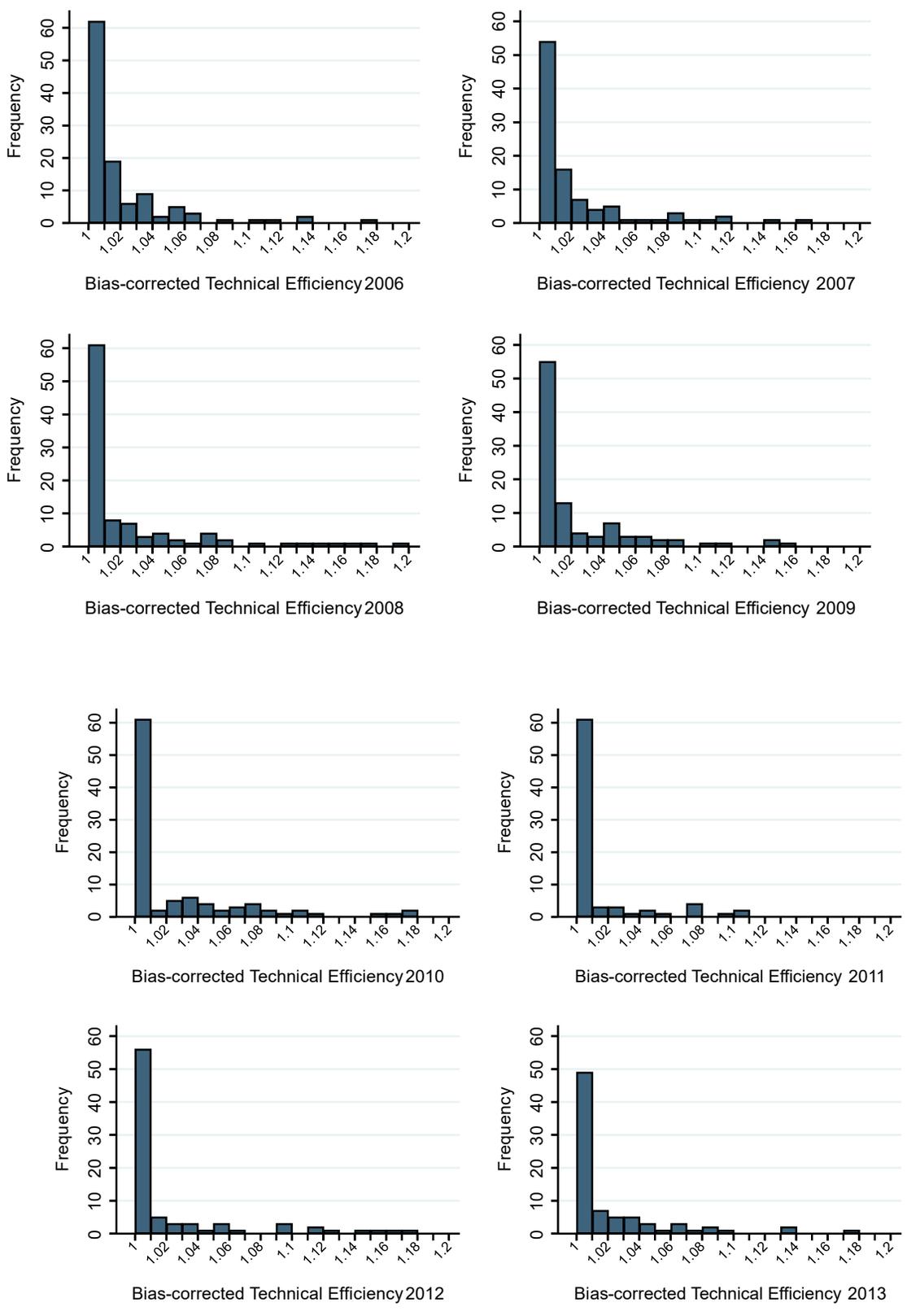


Figure E.1 Histogram of Technical Efficiency Estimates for Model A, 2006 – 2013

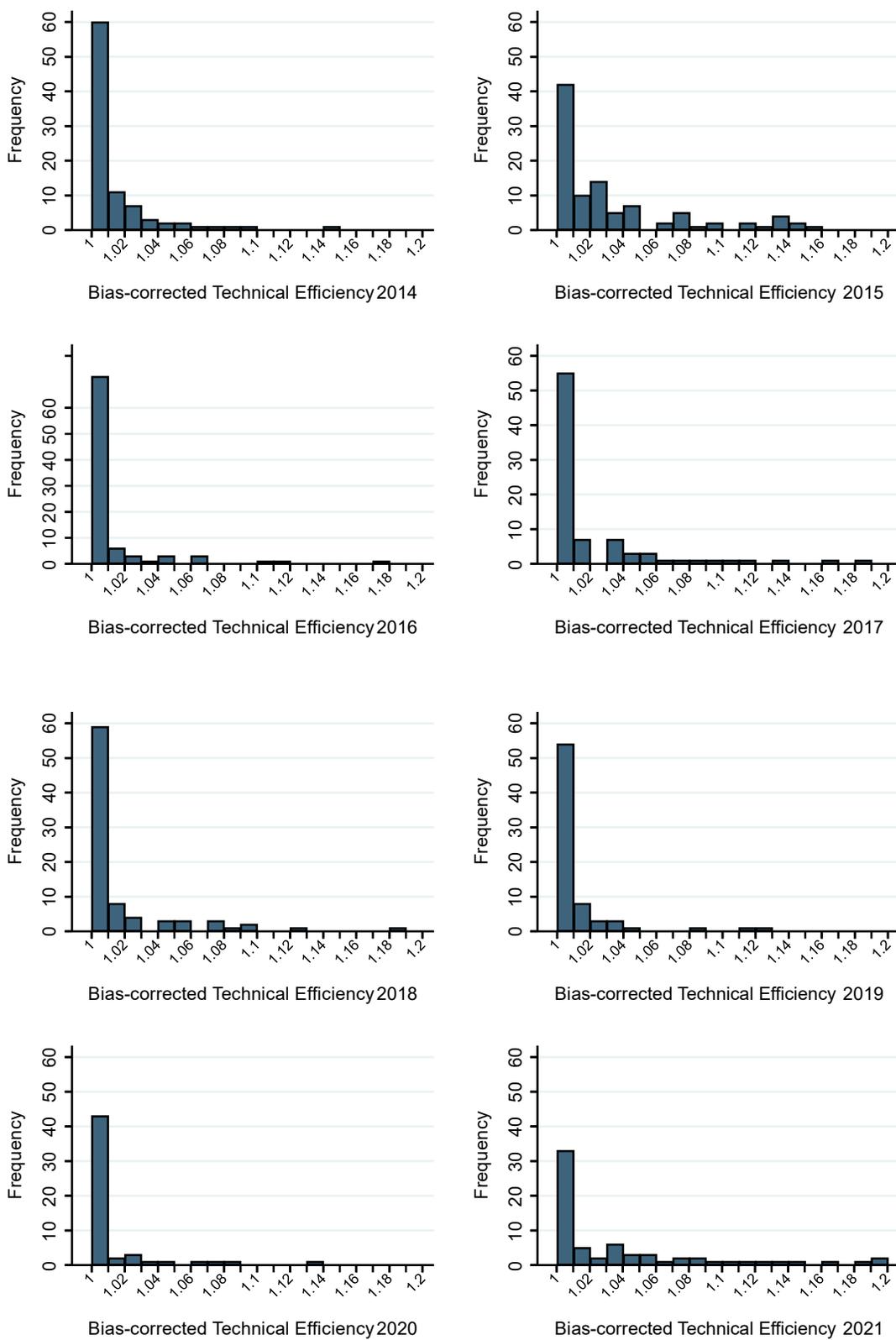


Figure E.2 Histogram of Technical Efficiency Estimates for Model A, 2014 – 2021

Appendix F: Effect of outliers in Data Envelopment Analysis

The discussions in Chapter 4 relies on the output-space to explain the production economics framework used in our study. In this section, we switch to the input-output space to discuss the effect of outliers and measurement errors. As mentioned, frontiers constructed using envelopment are sensitive to the dataset but are especially sensitive to outliers. Outliers in the data can arise from measurement errors or noise. Figure F.1 presents the theoretical frontier and DEA frontier for production in a slightly different manner. The solid black line is the theoretical production frontier in the input-output space. Similar to the output-space, the theoretical production frontier is defined as the boundary of the set of all feasible convex production (input-output) combinations. Thus, in the input-output space, it is a concave function defining the boundary of production sets. The production function follows the same assumptions but is just depicted in a different space.

In this example, the true theoretical frontier is represented using the solid black line. The researcher's observed dataset is made up of the red points: R, S, U, V, and W. Point S has some measurement error that leads to it being outside the theoretical feasible set. The observer cannot explicitly notice the influence of point S and the construction of the DEA frontier (dotted black line) is affected. The outlier causes the estimated frontier to cross the true production frontier. As a result, technical efficiency will be estimated incorrectly.

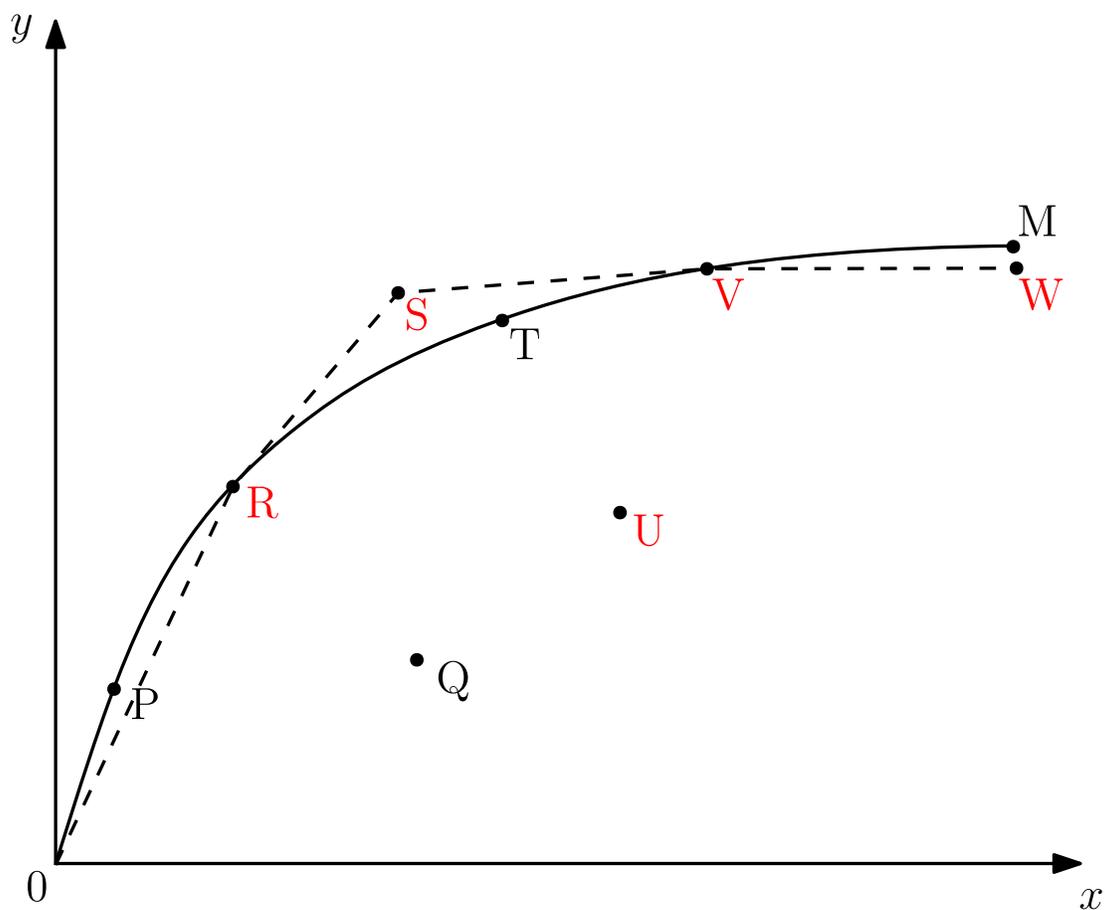


Figure F.1 Production Frontier and Outliers in Data

One approach for finding outliers in data was proposed by Stosic and Sousa (2003). Sousa and Stosic (2005) examine and estimate DEA scores of technical efficiency for a few thousand municipalities in Brazil. The interesting aspect of their research is the preparation of their data using a unique resampling technique. The aim of this technique is to reduce the influence of outliers and errors in the estimation of technical efficiency. The authors find that approaches for dealing with outliers usually rely on heavy manual inspection of the data. Therefore, to detect outliers in large databases, the work of Stosic and Sousa (2003) introduces a combination of bootstrapping and jackknifing, referred to as the Jackstrap.

The Jackstrap approach resamples data to detect outliers automatically. It aids in making non-parametric estimates more robust. The general working of the Jackstrap is based on calculating the effect of removing one production unit on the estimates of technical efficiency of all other production units. The authors refer to the measure of influence of the production unit held out on the technical efficiency scores of other units as 'leverage'. Leverage values are calculated for each production unit. Then, technical efficiency scores are calculated after dropping production units with the highest average leverage values. Their study concludes that the technical efficiency scores obtained from this approach provide more robust estimates. There is an opportunity to account for outliers by applying Jackstrap in our study after obtaining more observations to form a larger database.

Appendix G: Map of Weather Stations

Figure G.1 shows the location of weather stations used by ACIS to collect meteorological data. The yellow dots are the ACIS weather stations. The pink dot (located at the lower right-hand side of the photo) is a weather station that was currently selected on ACIS' web interface when the screen capture was taken.

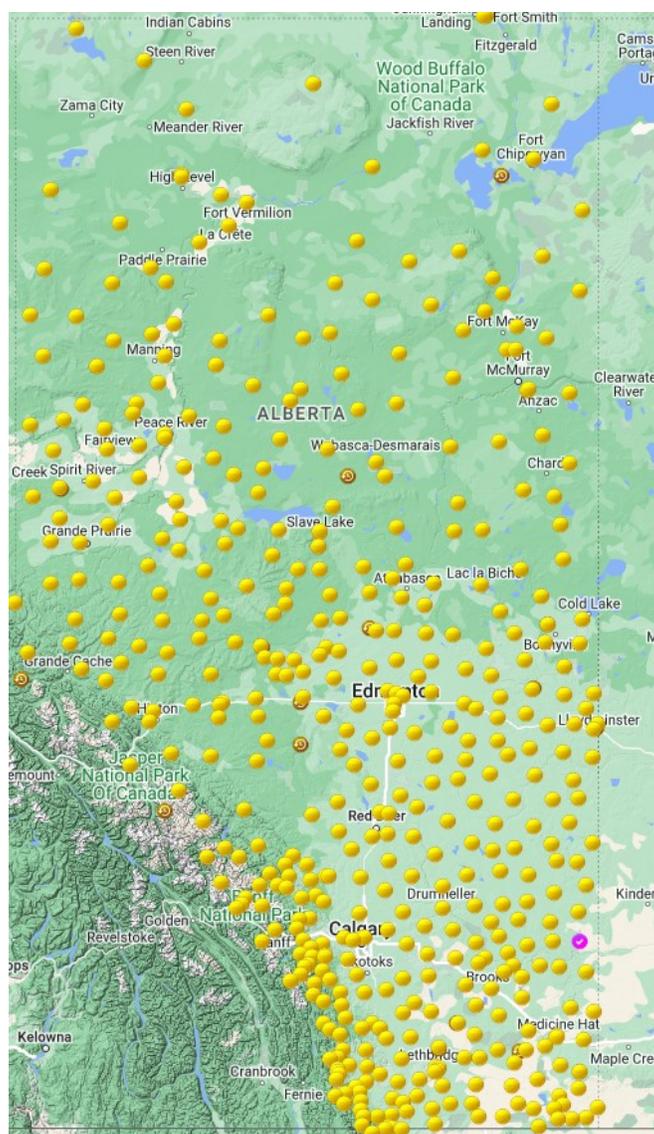


Figure G.1 Map of Weather Stations in Alberta
(Source: Alberta Climate Information Service 2020)

Appendix H: ACIS' Estimation for Modelling Incoming Radiation

Some weather stations are not equipped with sensors to measure incoming solar radiation. For these stations, the value for incoming radiation is estimated using the equations mentioned below. The modelled values show an 87% agreement between measured and estimated daily radiation observations (ACIS 2019).

First, data on the following variables is required:

G_{sc}	=	Solar constant ($MJm^{-2}day^{-1}$)
dfr	=	Inverse relative distance
$Jday$	=	Julian day
$Ndays$	=	Number of days in a year (366 in a leap year)
nws	=	Sunset hour angle
$ndec$	=	Solar declination angle
$elev$	=	Station elevation (meters)
lat	=	Station latitude (radians)

Second, the following equations are applied to calculate three intermediate variables that are needed to calculate an estimate for the extraterrestrial radiation:

Inverse relative distance given is by:

$$dfr = 1 + 0.033 \cdot \cos\left(\frac{2\pi \cdot Jdays}{Ndays}\right),$$

the sunset hour angle is calculated as:

$$nws = \arccos(-\tan(lat) \cdot \tan(ndec)),$$

and the solar declination angle is found using the following equation:

$$n_{dec} = 0.409 \cdot \sin\left(\frac{2\pi \cdot J_{days}}{N_{days}} - 1.39\right).$$

Using these three calculated variables along with the other variables mentioned earlier, the estimate for the extraterrestrial radiation (Ra) is calculated as:

$$Ra = \frac{1440}{\pi} G_{sc} \cdot drf(n_{ws} \cdot \sin(lat) \cdot \sin(n_{dec}) + \cos(lat) \cdot \cos(n_{dec}) \cdot \sin(n_{ws})).$$

Finally, the value for the estimated solar radiation (Rs) is given by applying Ra in the following equation:

$$Rs = Kt \times Ra \times (T_{max} - T_{min})^{0.5}.$$

Where:

Rs	=	Estimated solar radiation
Ra	=	Extraterrestrial radiation
T_{max}	=	Daily maximum air temperature
T_{min}	=	Daily minimum air temperature
Kt	=	Adjustment coefficient (equal to 0.16)

Appendix I: Machine Learning Model Specific Hyperparameters

Table I.1 provides the model specific hyperparameters and their tuned values that maximize the smoothed AUROC curve for the given ML model. The definition of each hyperparameter is found in the various R packages written by: Venables and Ripley (2002), Karatzoglou et al. (2004), Weihs et al. (2005), Alfaro, Gámez and García (2013), Schliep and Hechenbichler (2016), Wright and Ziegler (2017), Tuszynski (2021), Therneau and Atkinson (2022), Chen et al. (2023), Karatzoglou, Smola and Hornik (2023). These works developed different machine learning packages and are collectively applied using the *caret* package in RStudio (Kuhn et al. 2023).

Table I.1 Model-Specific Hyperparameters

Model	Hyperparameter
Logistic Regression	nIter (100) An integer, describing the total number of iterations for boosting or the number of decision stumps to use. Decision stumps are one node decision trees.
	fL (0) A factor for Laplace correction. The default factor is 0 (no correction).
Naïve Bayes	Usekernel (TRUE) A logical parameter. If true, a kernel density estimate is used for estimating the density. A normal density is estimated if set to false.
	adjust (1) An integer for bandwidth adjustment. It is an input for Kernel Density Estimation. The bandwidth in KDE represents the amount of spread in the KDE kernel function.
Neural Network	Size (20) Number of units in the hidden layer. By default, the model fits a neural network with one hidden layer.
	Decay (0.000128) Weight decay parameter that uses the sum off squares of the weights as a penalty. This helps with the optimization process and to avoid over-fitting.

Decision Trees	<p>Cp (0) The role of the complexity parameter is to save computing time by pruning off splits that are not worthwhile. It informs the program that any split that does not improve the model fit by the value of cp, will be pruned off.</p>
	<hr/> <p>Kmax (25) Maximum number of neighbors.</p>
K – Nearest Neighbor	<p>Distance (1.439) Minkowski distance parameter. The distance is measured between two points in N-dimensional space.</p> <p>Kernel (Inverse) Kernel type to use in the program. Options include “rectangular”, “triangular”, “epanechnikov”, “biweight”, “tri-weight”, “cos”, “inverse”, “gaussian”, and “optimal”.</p>
Support Vector Machines	<hr/> <p>Sigma (1.640) The sigma parameter determines how much a training example influences the decision boundary. For lower values of sigma, the <i>reach</i> is further. A high value of sigma forces the SVM boundary to be dependent on points that are closest to the decision boundary (ignoring points further away). Conversely, lower values of sigma will construct a boundary that accounts for points further away. Higher values lead to ‘flexed’ decision boundaries; lower values lead to a boundary that is more linear.</p> <p>Cost (914.342) This parameter is the tradeoff between misclassification of training examples and the simplicity of the decision boundary. Lower values lead to a smoother decision surface while higher values aim to classify all training points appropriately.</p>
	<hr/> <p>Mtry (5) The number of randomly selected predictors to split at each node. Default value is the square root of the total number of predictors.</p>
Random Forest	<p>Splitrule (gini) Splitting rule chosen for how cut-off points are picked. This parameter for classification and probability models can take the following values: ‘extratrees’, ‘gini’ (default), and ‘hellinger’.</p> <p>Minimum nodesize (1) Minimal node size with a default value of 1. This parameter affects how the decision tree is constructed. It is the number of instances in the last node. If splitting a node into two nodes results in one of them being smaller than the value of this parameter, the node is not split (making it a leaf node). It is a stopping criterion for the depth of the decision tree.</p>

Nrounds (816)

Number of iterations, in classification models, it is the number of trees to grow. The default value is 100.

Max.depth (10)

Maximum depth of a tree. The default value is 6.

Eta (0.265)

Shrinkage parameter controlling the learning rate. After each round, the feature weights are shrunk to reach the optimal results. Lower values cause slower computation and should be supported by a higher number of iterations. Default value of 0.3 with a range of (0,1).

Gamma (0.602)

A minimum loss reduction is needed before making an additional split on the leaf node of the tree. Higher values lead to conservative models. The default value is 0 and this parameter has a range of (0, inf).

**eXtreme
Gradient
Boosting
(XGBoost)****Subsample (0.818)**

Subsample percentage controlling the number of observations selected for constructing a tree. For instance, a value of 0.5 randomly collects half the data to grow trees with an aim to avoid overfitting. The default value is 1 and has a range of (0,1).

Colsample_bytree (0.533)

Subsample ratio of columns. The number of predictors supplied to construct a tree. Default value of 1 with a range of (0,1).

Rate_drop (0.304)

Fraction of trees dropped. Range of (0,1).

Skip_drop (0.799)

The probability of skipping a dropout. Range of (0,1).

Min_child_weight (1)

Minimum sum of instance weights required in a child. Default value of 1 with a larger value leading to a conservative model.

Mfinal (99)

Number of iterations for boosting or the number of trees to grow. The default value is 100.

**Adaptive
Boosting****Maxdepth (21)**

Maximum tree depth. It is a stopping criterion for how trees are constructed.

Coefflearn (Freund)

Coefficient for the learning method to use. Options include “Breiman”, “Freund”, “Zhu”. The default is Breiman.

Appendix J: Performance Measures for Machine Learning Models

Table J.1 provides model output metrics for all the machine learning models implemented in the study. The definitions of these metrics and descriptions are discussed by various authors and websites such as: the machine learning foundational courses by Google Developers (2023), machine learning textbooks authored by Hastie et al. (2001), Bishop (2006), Pedregosa et al. (2011), James et al. (2013), and Alpaydin (2020).

Root Mean Square Error (RMSE) measures the average magnitude of errors between predicted and actual values. Lower RMSE signifies better accuracy and model predictions are closer to the actual values. As seen in Table K.1, RMSE scores are lower for models that perform better than others, for instance, the RMSE for Logit is 0.561 (AUROC of 70.236%) while the RMSE for KNN is 0.281 (AUROC of 98.407%). Accuracy is calculated the proportion of correctly classified instances out of the total number of instances, a higher value of accuracy is desired. Kappa measures the agreement between predicted and actual classifications by considering that agreement can happen by chance. A high kappa value signifies that there is actual agreement between predicted and actual classes, and it is not just by chance. Kappa values indicate that models of KNN (0.834), RF (0.827) and AdaBoost (0.791) are much better than random classifiers in predicting the binary classes correctly. While models with a lower kappa value such as Logit (0.294) have a lot of room for improvement.

The F1 statistic is the harmonic mean of precision and recall (sensitivity). It balances the trade-off between precision and recall. Higher F1 scores are preferred as they indicate better overall performance by considering both the metrics of precision and recall. The models of KNN, RF, and

XGBoost show high F1 scores of 0.936, 0.933, and 0.899 respectively. Precision is defined as the proportion of correctly predicted positive instances out of all positive instances. A higher value of precision translates to a lower false positive rate (this is desirable if false positives are costly). Negative Predictive Value is proportion of correctly predicted negative instances out of all negative instances. Higher NPV values indicate a lower false negative rate, which is desirable especially in situations where false negatives are costly. The better performing models from the nine models we develop show higher values of NPV.

Sensitivity (recall) or true positive rate is the proportion of correctly predicted positive instances out of all actual positive instances. Higher sensitivity values indicate a lower false negative rate (desirable if false negatives are costly). Specificity is the true negative rate that measures the proportion of correctly predicted negative instances out of all actual negative instances. Higher specificity values indicate a lower false positive rate, which is desirable. Prevalence is the proportion of positive instances in the dataset. It is useful for understanding the distribution of the target variable if needed.

Detection rate or true positive rate measures the proportion of correctly predicted positive instances out of all actual positive instances. Higher detection rate signals a stronger, more powerful model. Detection prevalence measures the proportion of instances predicted as positive out of the total number of instances. It is useful for understanding the distribution of the predicted positive class. Finally, balanced Accuracy calculates the average of sensitivity and specificity. It provides an overall measure of the model's performance in both positive and negative instances.

Table J.1 Output Metrics for ML Models

Model Metrics	Machine Learning Models								
	Logit	NB	NN	TREE	KNN	SVM	RF	XGBoost	AdaBoost
RMSE	0.561	0.558	0.457	0.497	0.281	0.628	0.287	0.351	0.315
Accuracy	0.685	0.688	0.791	0.753	0.921	0.606	0.918	0.877	0.901
Kappa	0.294	0.349	0.558	0.478	0.834	0.040	0.827	0.741	0.791
F1-score	0.772	0.742	0.832	0.801	0.936	0.751	0.933	0.899	0.919
Precision (Pos. Pred. Value)	0.675	0.728	0.795	0.767	0.903	0.601	0.903	0.874	0.887
Neg. Pred. Value	0.721	0.625	0.784	0.728	0.953	1.000	0.944	0.881	0.925
Sensitivity	0.902	0.757	0.873	0.838	0.971	1.000	0.965	0.925	0.954
Specificity	0.370	0.588	0.672	0.630	0.849	0.034	0.849	0.807	0.824
Prevalence	0.592	0.592	0.592	0.592	0.592	0.592	0.592	0.592	0.592
Detection Rate	0.534	0.449	0.517	0.497	0.575	0.592	0.572	0.548	0.565
Detection Prevalence	0.791	0.616	0.651	0.647	0.637	0.986	0.634	0.627	0.637
Balanced Accuracy	0.636	0.673	0.773	0.734	0.910	0.517	0.907	0.866	0.889
AUROC	0.702	0.710	0.819	0.799	0.984	0.924	0.954	0.918	0.941

Appendix K: Feature Importance

Feature importance values are presented for each model in Table K.1. Table K.2 supports the graph in Figure 6.4 and shows the values of the AUROC Weighted Permutation Feature Importance for the six predictors used in our machine learning models.

Table K.1 Model-specific Permutation Feature Importance

Feature	Machine Learning Modles								
	Logit	NB	NN	TREE	KNN	SVM	RF	XGBoost	AdaBoost
Air Temp.	0.009	0.008	0.184	0.042	0.230	0.287	0.028	0.031	0.073
Humidity	0.084	0.066	0.239	0.212	0.249	0.291	0.119	0.060	0.093
Inc. Radiation	0.078	0.104	0.202	0.128	0.277	0.307	0.067	0.043	0.041
Precipitation	0.045	0.010	0.124	0.056	0.164	0.204	0.035	0.034	0.034
Snow Water	0.067	0.045	0.107	0.160	0.150	0.136	0.039	0.037	0.029
Wind Speed	0.028	0.035	0.212	0.130	0.241	0.294	0.075	0.089	0.058

Table K.2 AUROC Weighted Feature Importance

Feature	AUROC Weighted Permutation Feature Importance
Humidity	0.189
Inc. Radiation	0.184
Wind Speed	0.171
Air Temp.	0.151
Precipitation	0.110
Snow Water	0.103

Figure K.1 plots the nine machine learning models based on the weights assigned to them. The dotted trendline shows the relationship between the AUROC values of the models and their assigned weights. Models with higher AUROC values are more powerful, therefore, they are given

a higher weight such as KNN, RF, and AdaBoost. These models have more influence on the final feature importance as compared to models that are not as powerful such as Logit and NB.

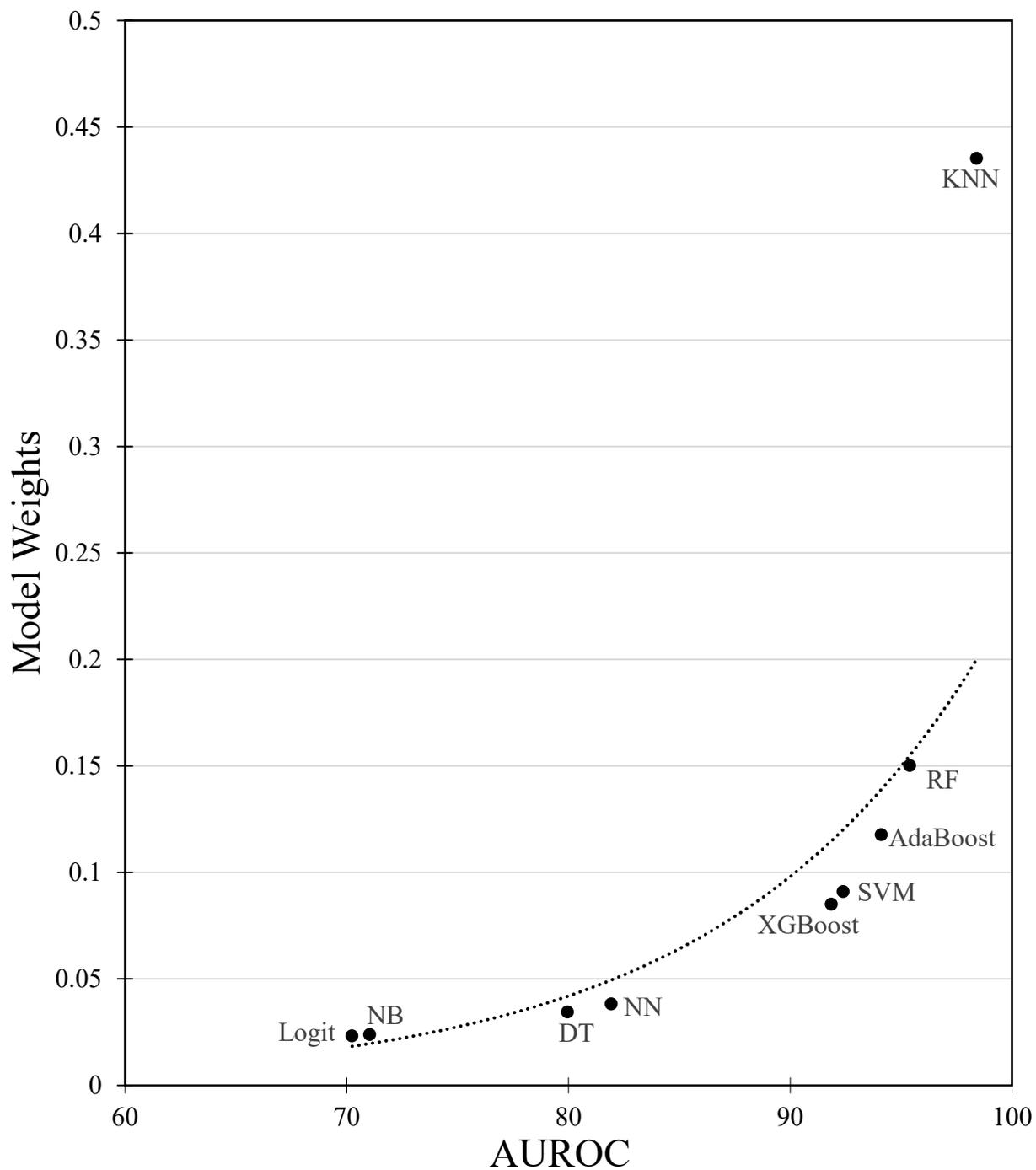


Figure K.1 Model Weights and AUROC

Appendix L: Robustness Check for AUROCs and Feature Importance

The criterion that a lookout tower is deemed technically efficient if the bias-corrected technical efficiency is less than or equal to 1.01 and not technically efficient otherwise is used as the criterion to construct Figures 5.2 and 5.3 as well as for the OLS regressions shown in Table 5.3. In this section of the Appendix, we use the remaining criteria that are mentioned in Chapter 5, Table 5.2. In total, we choose five criteria for classifying a lookout tower as technically efficient ($\theta \leq 1.005, 1.010, 1.015, 1.020, 1.050$). This Appendix provides AUROC scores for all the machine learning models as well as the AUROC Weighted Permutation Feature Importance values based on each of these five criteria. Table M.1 provides the models' AUROC scores for each one of the classification criteria.

Table L.1 AUROC Values for Different Classification Criteria

ML Model	AUROC for different criteria				
	1.005	1.01	1.015	1.02	1.05
KNN	97.370	98.407	97.869	97.664	97.722
AdaBoost	96.064	94.108	96.312	96.562	94.990
RF	95.839	95.383	96.304	95.666	95.570
XGBoost	94.256	91.849	94.749	93.095	92.114
SVM	91.936	92.388	88.010	90.492	88.301
NN	74.651	81.930	77.344	80.523	71.876
DT	76.633	79.949	77.530	78.210	70.349
NB	63.849	71.035	65.423	67.354	58.857
Logit	68.388	70.236	70.336	72.322	75.643

Table L.2 tabulates the same rule of thumb we used in the results in section 6.4. A ML model's ability to predict is labelled 'great' if the AUROC is more than 90 and 'good' if the AUROC is more than 80 (Adamecz-Völgyi, Henderson and Shure 2020). We add to this by

labelling a model's AUROC as 'unsatisfactory' if it is below 80 and 'poor' if the AUROC is below 70. We use the term performance category to refer to these labels.

Table L.2 Performance Categories for Classifying AUROC Values

Inequality	Performance Category
$\text{AUROC} \geq 90$	Great (Gr)
$90 > \text{AUROC} \geq 80$	Good (Gd)
$80 > \text{AUROC} \geq 70$	Unsatisfactory (U)
$\text{AUROC} < 70$	Poor (P)

Table L.3 provides the performance category for the models at each classification. K-Nearest Neighbors, Adaptive Boosting, Random Forests, and eXtreme Gradient Boosting have AUROC values of more than 90 in all the classification criteria. Support Vector Machines falls under either great or good categories. Neural Network performs unsatisfactory for the 1.005, 1.015, and 1.05 criteria but good for the 1.01 and 1.02 classification criteria. Decision Tree performs at an unsatisfactory level for all criteria while Naïve Bayes' and Logistic Regression models are either unsatisfactory or poor in their AUROC values.

Table L.3 AUROC Performance Categories for Different Classification Criteria

ML Model	AUROC				
	$\theta \leq 1.005$	$\theta \leq 1.01$	$\theta \leq 1.015$	$\theta \leq 1.02$	$\theta \leq 1.05$
KNN	Gr	Gr	Gr	Gr	Gr
AdaBoost	Gr	Gr	Gr	Gr	Gr
RF	Gr	Gr	Gr	Gr	Gr
XGBoost	Gr	Gr	Gr	Gr	Gr
SVM	Gr	Gr	Gd	Gr	Gd
NN	U	Gd	U	Gd	U
DT	U	U	U	U	U
NB	P	U	P	P	P
Logit	P	U	U	U	U

Table L.4 provides the AUROC Weighted Permutation Feature Importance for the weather variables across the different classification criteria. Across all classification criteria, humidity is

the most influential with the highest AUROC Weighted Permutation Feature Importance value. Incoming radiation and wind speed place second or third based on the criteria we look at. For instance, for criteria 1.005, 1.01, and 1.05, incoming radiation is more influential than wind speed, but it is the other way around for criteria 1.015 and 1.02. Air temperature is the fourth most influential weather variable in predicting the technical efficiency class of lookout towers across all five criteria except criteria 1.005 (in which it is ranked second). Precipitation and snow water are the two least influential predictors across the five criteria.

Table L.4 Feature Importance across Different Classification Criteria

Feature	AUROC Weighted Permutation Feature Importance on Different Criteria				
	$\theta \geq 1.005$	$\theta \geq 1.01$	$\theta \geq 1.015$	$\theta \geq 1.02$	$\theta \geq 1.05$
Humidity	0.205	0.189	0.184	0.200	0.230
Incoming Radiation	0.155	0.184	0.156	0.138	0.175
Wind speed	0.146	0.171	0.166	0.148	0.167
Air temperature	0.154	0.151	0.144	0.128	0.156
Precipitation	0.103	0.110	0.090	0.088	0.112
Snow water	0.127	0.103	0.095	0.102	0.096

Appendix M: Short-run versus Long-run Production Functions

Wildfire detection requires the utilization of some inputs to produce detection outputs via a production process. Assume that a lookout tower produces wildfire detection, and this process is defined by a production function, f , which we assume uses two inputs: visibility profile V , and other inputs, I (e.g., lookout observer experience). The output, Q , can be expressed as a function of inputs as:

$$Q = f(I, V) \quad (11)$$

Production function in fundamental economic theory can be categorized as short-run production or long-run production. The short run is defined as the situation where at least one of the inputs that goes into the production process is fixed. In our case, the visibility profile is a fixed input used in the production of wildfire detections. Figure N.1 provides a graph of how detection output varies with the level of I . The black line provides a production function that produces output Q_1 based on different levels of other inputs when holding visibility fixed at V_1 . Any movement along the production function shows that output increases (following diminishing marginal rate of return) as the level of other inputs increases given the fixed amount of visibility. This formulation of the production function is said to be a short-run production function. The function can shift upwards or downwards only if the fixed input (visibility profile) can vary. For instance, the dashed line provides a detection production function that shows how output changes based on different levels of other inputs if visibility is now fixed at V_2 .

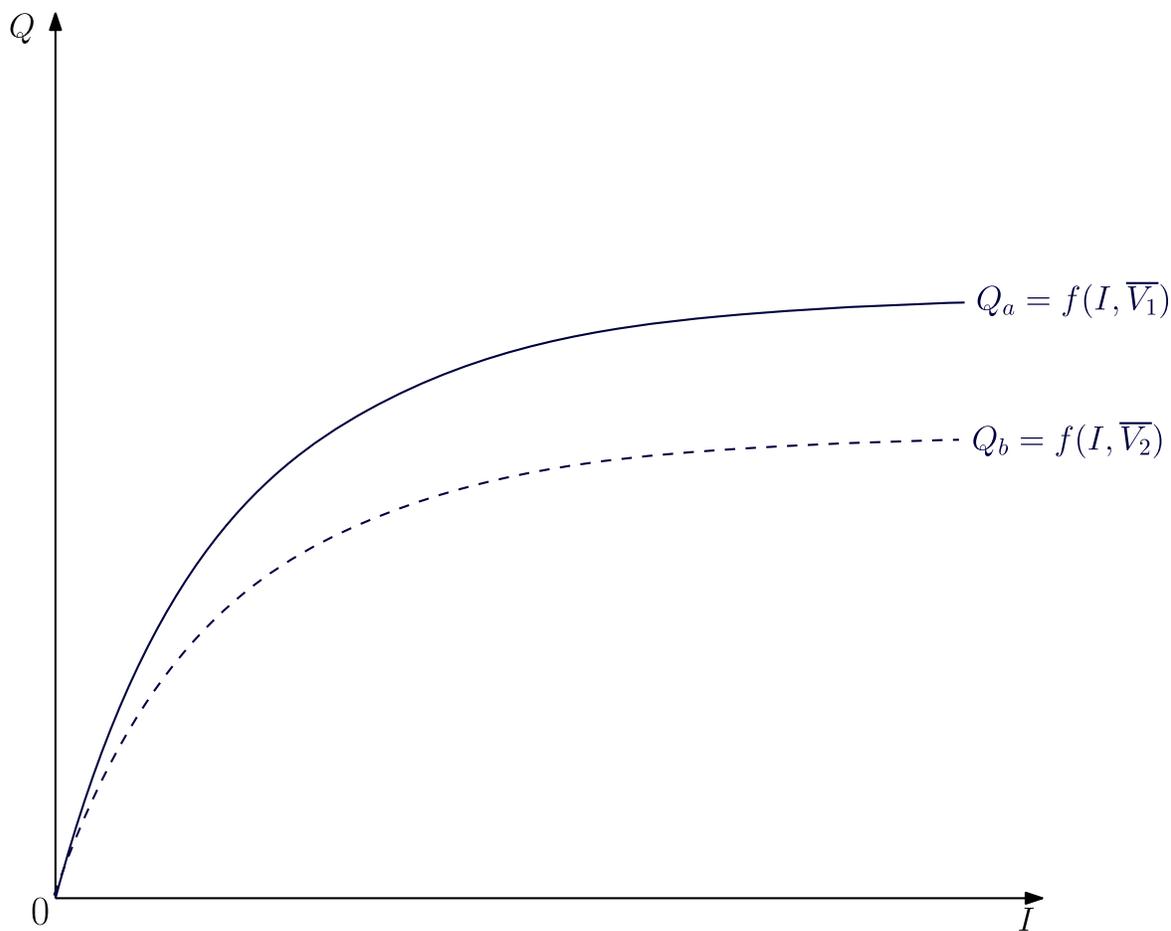


Figure M.1 Short-run Production Function

In contrast, the notion of long run production relies on none of the inputs being fixed, i.e., visibility and all other inputs can vary over time. The lookout tower now utilizes a mix of the inputs. They use varying quantities of both inputs (as compared to only choosing the level of other inputs when visibility is fixed) to produce a certain level of detection output. In this scenario, output changes as the combinations of inputs used change. To depict this graphically, we use isoquants. An isoquant is a graphical representation of all different combinations of inputs that can be used in the production of a specific level of detection output by a lookout tower. While it is possible to represent the production function using a general mathematical function such as equation (11), plotting this graphically can be challenging when dealing with multiple inputs of varying levels.

Isoquants help provide a more informative visualization of the production process by showing all various combinations of inputs that produce a feasible level of output. In our example, the lookout can use more of one input and less of another to produce the same output.

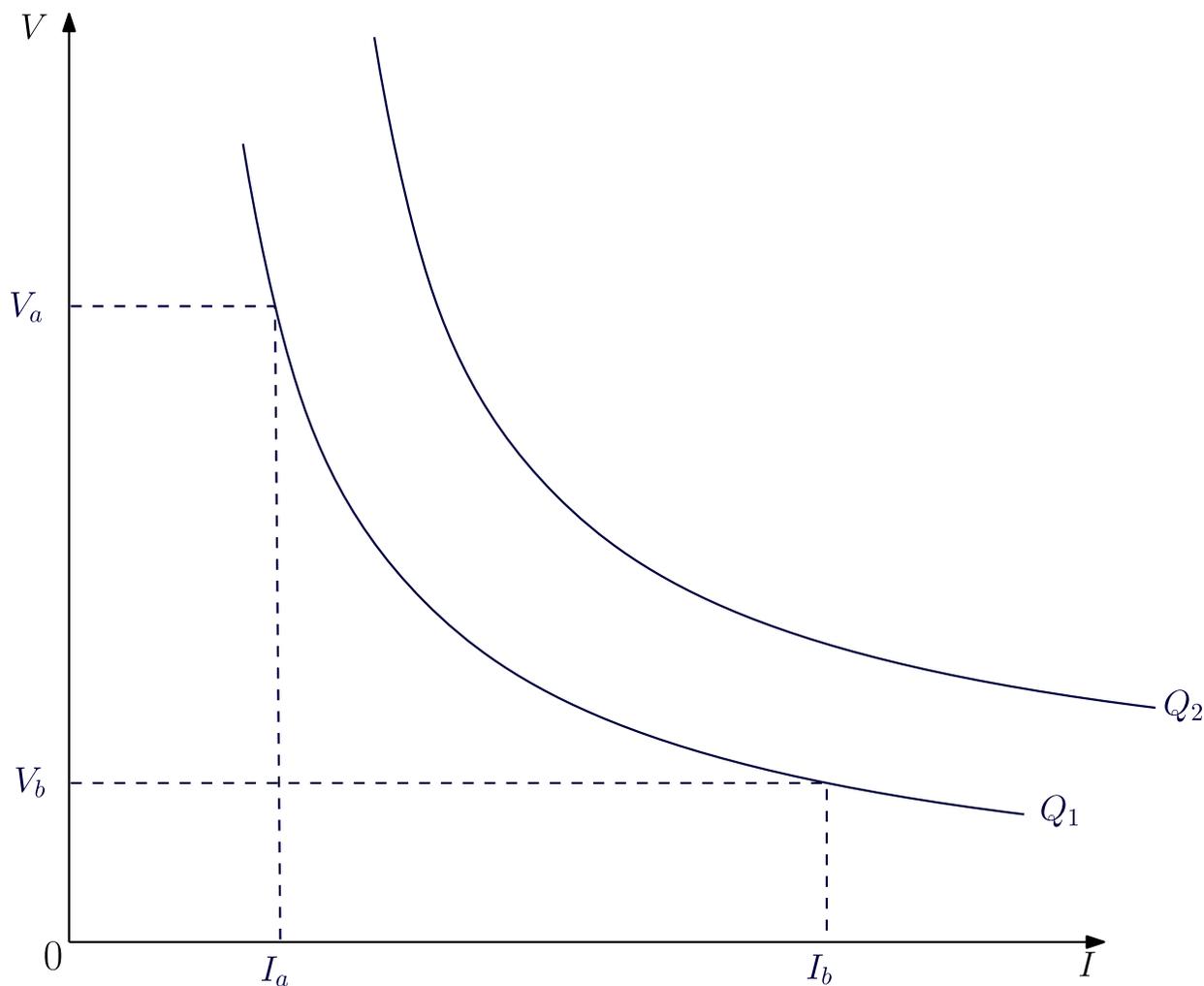


Figure M.2 Long-run Production Function

Figure N.2 includes two isoquants for two production levels: Q_1 and Q_2 as shown. The first isoquant depicts all input combinations of visibility and other inputs that can produce Q_1 . For example, both the combinations (I_a, V_a) and (I_b, V_b) produce the same amount of detection output (Q_1). Additionally, with varying input levels, isoquants can shift upwards (or downwards) based

on factors such as technological improvements for efficient production, increased labor skills, or placement of the lookout tower (changing the visibility), all leading to a lookout tower being able to increase productivity in this example. The effect of this would cause the isoquant to shift upwards as shown by the second isoquant which now produces more output at level Q_2 . As such, the scenario where input levels can change over time and vary output levels, is considered as long run production.

The realization of lookout tower production used in our analysis is in the short run. The inputs (area directly and indirectly visible) are fixed and do not change over time. Assume a scenario where the decision maker can decide to easily relocate lookout towers. This changes the inputs that lookout towers use and therefore provides a long run perspective of lookout towers' production process. As discussed in section 7.1, the addition of time-varying data would transition the realization of the detection production function from a short run approach and closer to the long run. As such, it will provide a different perspective on the production of wildfire detection by lookout towers over time.

Appendix N: Code Availability and Replication Guide for OSF

We have created an Open Science Framework (OSF) repository that provides all necessary data files, working directory file structure, and codes to fully replicate all results presented in this study.

The link to the OSF repository is: <https://osf.io/ywpxf/>.

Before starting the replication process, be aware that:

- The empirical work was performed using two software: Stata 17 and R 4.2.1.
- The R codes provided use parallel processing and the code detects and uses the maximum number of cores of your machine.
- R codes (in the folder entitled “Codes”) include “library()” and "install.packages()" commands. The "install.packages()" code lines need to be uncommented if your machine does not have the required packages. Similarly, the Stata do file (entitled “5_Figures and Tables”, in “Codes”) requires two packages (tabout and logout) that need to be installed. If your Stata does not have these packages, please uncomment the “ssc install” commands and execute them.

Please follow the instructions below to replicate all results.

1. Download the zip file labelled “Replication” from the OSF repository and extract all items.

The directory structure of the folder “Replication”, and the associated files, are as follows:

Replication

- Auxiliary
 - Permutation Importance
- Codes
 - 1_Technical_Efficiency.R
 - 2_ML_Random_Search_and_Splines.R
 - 3_ML_Tuned_Models.R

- 4_ML_Robustness_Checks.R
- 5_Figures_and_Tables.do
- Figures
- Raw Data
 - DEA_Dataset.dta
 - Fire_Database.dta
 - Monthly_Weather.dta
 - ML_Dataset.dta
 - ML_Dataset1.dta
 - ML_Dataset2.dta
 - ML_Dataset3.dta
 - ML_Dataset4.dta
 - TE_Database.dta
 - Tower_Database.dta
- Results
 - Random Search Results
 - TE Results
- Tables

Note: The thesis uses Machine Learning to classify towers as Technically Efficient or not. The file “ML_Dataset.dta” is used in the main analysis where towers are classified as technically efficient if their bias-corrected efficiency score $\theta \leq 1.01$. The robustness checks in Appendix L use alternative classification criteria, namely:

- $\theta \leq 1.02$ (calls the data file “ML_Dataset1.dta”),
 - $\theta \leq 1.05$ (calls the data file “ML_Dataset2.dta”),
 - $\theta \leq 1.005$ (calls the data file “ML_Dataset3.dta”), and
 - $\theta \leq 1.015$ (calls the data file “ML_Dataset4.dta”).
2. Open the folder “~/Replication/Codes” and execute the four R codes in the order below:
- 1_Technical_Efficiency.R
 - 2_ML_Random_Search_and_Splines.R
 - 3_ML_Tuned_Models.R
 - 4_ML_Robustness_Checks.R

Note: the user must set the working directory in each R file, i.e. the path to the "Replication" folder (excluding the "Replication" folder).

- The computational times for the R codes are provided in the table below. Note that the times are based on a machine with the following specifications: 128GB RAM, AMD Ryzen 9 3950X 16-core processor 3.49 GHz, and Windows 10 Enterprise as the OS.

File name	Computing time
1_Technical_Efficiency.R	~ 1 min
2_ML_Random_Search_and_Splines.R	~ 35 hrs
3_ML_Tuned_Models.R	~ 2 hrs 30 mins
4_ML_Robustness_Checks.R	~ 13 hrs 30 mins

- Open "~/Replication/Codes" and execute the do-file "5_Figures_and_Tables.do". The table below shows the computing time in our machine.

Note: the user again must set the working directory, i.e. the path to the "Replication" folder (excluding the "Replication" folder).

File name	Computing time
5_Figures_and_Tables.do	~ 35 secs

The codes produce figures and tables and store them in their respective folders. The tables below link figures and tables to their source codes.

Figure number	Format	Code
3.2	.svg	5_Figures_and_Tables.do
3.3	.svg	5_Figures_and_Tables.do
3.4	.svg	5_Figures_and_Tables.do
4.4	.svg	1_Technical_Efficiency.R
4.5	.svg	1_Technical_Efficiency.R
5.1	.svg	1_Technical_Efficiency.R
5.2	.svg	5_Figures_and_Tables.do
6.3	.svg	3_ML_Tuned_Models.R
6.4	.svg	3_ML_Tuned_Models.R
6.5	.svg	3_ML_Tuned_Models.R
B.1	svg	5_Figures_and_Tables.do
E.1	.svg	5_Figures_and_Tables.do
K.1	.csv	3_ML_Tuned_Models.R

Table number	Format	Code
2.1	txt	5_Figures_and_Tables.do
2.2	txt	5_Figures_and_Tables.do
2.3	txt	5_Figures_and_Tables.do
3.1	csv	5_Figures_and_Tables.do
5.1	txt	5_Figures_and_Tables.do
5.2	csv	5_Figures_and_Tables.do
5.3	txt	5_Figures_and_Tables.do
J.1	csv	3_ML_Tuned_Models.R
K.1	csv	3_ML_Tuned_Models.R
K.2	csv	3_ML_Tuned_Models.R
L.1	csv	4_ML_Robustness_Checks.R
L.4	csv	4_ML_Robustness_Checks.R