

THOMPSON RIVERS UNIVERSITY

Advanced Wildfire Prediction with Machine Learning:
Leveraging Liver Cancer Algorithm and Spiral Updates for
Feature Selection.

By

MODAMORI OLUWAYOMI O

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Master of Science in Environmental Science

KAMLOOPS, BRITISH COLUMBIA

December, 2024

SUPERVISOR:

Dr. Mohamed Tawhid

CO-SUPERVISOR:

Dr. Emad Mohammed

© *Modamori Oluwayomi O, 2025*

ABSTRACT

Wildfires threaten ecosystems, economies, and public health, particularly in high-risk regions. Accurate wildfire prediction remains challenging due to complex interactions among weather patterns, vegetation dynamics, climate change, and human activities. This study investigates the role of advanced metaheuristic algorithms in optimizing feature selection for wildfire prediction across eight Canadian provinces, focusing on improving accuracy and computational efficiency. We evaluate twelve algorithms, including Atom Search Optimization (ASO), Barnacles Mating Optimizer (BMO), Chef-Based Optimization (CBO), Energy Valley Optimizer (EVO), Equilibrium Optimizer (EO), and Walrus Optimization Algorithm (WOA), among others.

Key results highlight the superior performance of BMO and EVO, with BMO achieving average recall rates of 77.55% in Alberta and 76.51% in Quebec, and EVO attaining 76.96% and 78.30% in these provinces, respectively. In contrast, ASO consistently underperformed, yielding recall rates as low as 44.67% in Ontario and 51.99% in the Northwest Territories. Statistical analyses using Friedman and Wilcoxon signed-rank tests confirmed significant differences in algorithmic performance ($p < 0.05$), with spiral-enhanced variants of the Liver Cancer Algorithm (LCA) outperforming the baseline LCA. Furthermore, Random Forest and Gradient Boosting emerged as the most reliable prediction models, emphasizing the synergy between optimized feature selection and robust machine learning frameworks.

A significant contribution of this research is the enhancement of the LCA through spiral updates, specifically the Euler Spiral, which improves the balance between exploration and exploitation in the search space. This enhancement addresses the instability and slow convergence often associated with the standard LCA. Although the spiral updates improved LCA's performance, algorithms like

EVO and the Genetic Algorithm (GA) consistently outperformed LCA in recall and overall predictive accuracy across provinces.

The findings of this study highlight the variability in algorithm performance and emphasize the importance of tailoring wildfire prediction strategies to specific regional conditions. This work not only evaluates the capabilities of advanced metaheuristic algorithms but also identifies key environmental factors that predict wildfire risk, offering practical insights to enhance wildfire risk management and mitigation efforts.

Key Words: Wildfire; Machine Learning; Feature Selection; Model Optimization;
Liver Cancer Algorithm

ACKNOWLEDGEMENTS

Contents

1	Introduction	1
1.1	Background and Significance	1
2	Literature Review	7
2.1	Research History of Wildfire Science In Canada	8
2.1.1	Related Work In Machine Learning and Wildfire Science	11
3	Methodology	26
3.1	Study Framework	27
3.1.1	Objective Of The Study	28
3.1.2	Hypotheses and Research Questions	29
3.2	Data Collection and Sources	30
3.2.1	Data Types and Origin	30
3.2.2	Preprocessing Steps	34
3.3	Study Areas	40
3.4	Feature Selection and Metaheuristic Algorithm Implementation	43
3.4.1	Metaheuristic Algorithm Rationale	45
3.4.2	Algorithm Descriptions	48
3.4.3	Mathematical Model of LCA	79
3.4.4	Algorithm Mechanics	83
3.5	Proposed Algorithm	95

3.5.1	Spiral Implementation	95
3.6	Evaluation Metrics	129
3.6.1	Mean Fitness Function	129
3.6.2	Best Fitness Function	131
3.6.3	Worst Fitness Function	131
3.6.4	Standard Deviation	131
3.6.5	Classification Accuracy (CA)	132
3.6.6	Feature Selection Ratio (FSR)	132
3.6.7	F-score	132
3.6.8	Precision and Recall	133
3.7	Statistical Tests	133
3.7.1	Wilcoxon Signed-Rank Test	134
3.7.2	Friedman Test	134
3.7.3	Algorithm Performance Analysis	135
3.7.4	Regional Impact Analysis	135
3.8	Expected Outcomes and Limitations	136
3.9	Ethical and Practical Considerations	137
3.9.1	Ethical Data Use	137
3.9.2	Practical Implications	137
3.10	Summary of the Study Framework	137
4	Discussion	139
4.1	Overview of Study Objectives and Methodology	139
4.1.1	Provincial Results	140
4.2	Provincial Result Summary	166
4.2.1	Key Insights Across Provinces	169
5	Conclusion	172

5.1	Key Advancements	172
5.2	Environmental Variability	173
5.3	Computational Constraints	174
5.4	Future Work	174
5.5	Closing Remarks	176

List of Figures

3.1	Map of Study Area across Canadian Provinces	44
3.2	Visualization of the Archimedean Spiral	100
3.3	Visualization of the Euler Spiral	107
3.4	Visualization of the Fermat Spiral	111
3.5	Visualization of the Golden Spiral	116
3.6	Visualization of the Hyperbolic Spiral	120
3.7	Visualization of the Lituus Spiral	122
3.8	Visualization of the Logarithmic Spiral	128
4.1	Average Recall Ranks (Friedman Test)	144
4.2	Visualization of average recall rankings among algorithms in BC (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA. . .	147
4.3	Visualization of average recall rankings among algorithms in Mani- toba (Friedman Test results). Lower ranks indicate better perfor- mance, with several spiral-enhanced algorithms outperforming the LCA.	151
4.4	Visualization of average recall rankings among algorithms in North- west Territories (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA.	154

4.5	Visualization of average recall rankings among algorithms in Ontario (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA. . .	158
4.6	Visualization of average recall rankings among algorithms in Quebec (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA. . .	160
4.7	Visualization of average recall rankings among algorithms in Saskatchewan (Friedman Test results). Lower ranks indicate better performance, with spiral-enhanced algorithms generally outperforming the LCA.	163
4.8	Visualization of average recall rankings among algorithms in Yukon (Friedman Test results). Lower ranks indicate better performance, with spiral-enhanced algorithms demonstrating significant improvements.	167
4.9	Wilcoxon Signed-Rank Test Results vs. LCA	168

List of Tables

3.1	Summary of Features from FIRMS and ERA5 Data Sources	31
3.2	Comparison of Spiral Types for Feature Selection	130
4.1	Algorithm Rankings for Alberta	142
4.2	Spiral Algorithm Rankings for Alberta	143
4.3	Algorithm Rankings for British Columbia	146
4.4	Spiral Algorithm Rankings for British Columbia	146
4.5	Algorithm Rankings for Manitoba	149
4.6	Spiral Algorithm Rankings for Manitoba	150
4.7	Algorithm Rankings for Northwest Territories (NWT)	152
4.8	Spiral Algorithm Rankings for Northwest Territories (NWT)	153
4.9	Algorithm Average Recall and Rankings for Ontario	155
4.10	Algorithm Rankings for Ontario	157
4.11	Algorithm Rankings for Quebec	159
4.12	Spiral Algorithm Rankings for Quebec	160
4.13	Algorithm Rankings for Saskatchewan	161
4.14	Algorithm Rankings for Yorkton	164
4.15	Spiral Algorithm Rankings for Yorkton	166
4.16	Top 10 Most Selected Features by All Algorithms	171

List of Algorithms

1	Inputs and Outputs	49
1	Atom Search Optimization (ASO) for Feature Selection	50
2	Barnacles Mating Optimizer (BMO) for Feature Selection	53
3	Chef-Based Optimization Algorithm (CBOA) for Feature Selection .	55
4	Equilibrium Optimizer (EO) for Feature Selection	57
5	Exponential Distribution Optimizer (EDO) for Feature Selection . .	60
6	Energy Valley Optimizer (EVO) for Feature Selection	63
7	Genetic Algorithm (GA) for Feature Selection	65
8	Golden Ratio Method (GRM) for Feature Selection	68
9	Manta Ray Foraging Optimization (MRFO) for Feature Selection .	70
10	Particle Swarm Optimization (PSO) for Feature Selection	72
11	Walrus Optimization Algorithm (WAOA) for Feature Selection . . .	75
12	Liver Cancer Algorithm (LCA) for Feature Selection	85
13	Manta Ray Foraging Optimization (MRFO) for Feature Selection .	87
14	Particle Swarm Optimization (PSO) for Feature Selection	89
15	Walrus Optimization Algorithm (WAOA) for Feature Selection . . .	91
16	Liver Cancer Algorithm (LCA) with Archimedean Spiral Update. .	99
17	Liver Cancer Algorithm (LCA) with Detailed Euler Spiral Update for Feature Selection	104

18	Liver Cancer Algorithm (LCA) with Detailed Fermat Spiral Update for Feature Selection	108
19	Liver Cancer Algorithm (LCA) with Detailed Golden Spiral Update for Feature Selection	112
20	Liver Cancer Algorithm (LCA) with Hyperbolic Spiral Update for Feature Selection	117
21	Liver Cancer Algorithm (LCA) with Detailed Lituus Spiral Update for Feature Selection	121
22	Liver Cancer Algorithm (LCA) with Detailed Logarithmic Spiral Update for Feature Selection	125

List of Equations

1	Wind speed calculation based on vector components.	35
2	Calculation of wind direction in degrees.	36
3	Adjusted formula to align with 360° scale.	36
4	Relative humidity based on the August-Roche-Magnus approximation. .	36
5	Fitness Function for Feature Subset Evaluation	46
6	Feature Ratio Calculation	47

Chapter 1

Introduction

1.1 Background and Significance

Wildfires are critical and recurring natural disasters and have a profound and lasting influence on various aspects of life and the global environment. A wildfire is an uncontrolled fire spreading through vegetative fuels, exposing and consuming the natural and built environment in its path (Pyne et al. [1996]). While fire is a natural component of many ecosystems, facilitating plant regeneration and nutrient cycling (Santín and Doerr [2016]), its frequent and intense occurrence—often worsened by human activities and climate change has far-reaching consequences (Prapas et al. [2021]). Recent data highlights this escalating challenge: 2021 was notably one of the harshest years in recent forest fire history, with a staggering 9.3 million hectares of tree cover lost globally, a figure representing over a third of all tree cover loss that year (Tyukavina et al. [2022], Potapov et al. [2017]). In 2022, Canada saw a decline in wildfire activity, yet the situation remained concerning, with over 6.6 million hectares lost to fires. By 2023, the severity of wildfires intensified dramatically, as

nearly 15 million hectares—an area proximate to the size of Portugal were consumed (Weisse et al. [2022], Canada [2024]). This trend continued into 2024, with over 5.3 million hectares burned by September (CIFFCI [2024]). This activity significantly contributed to the year’s global wildfire-induced carbon emissions, estimated at around 2170 megatonnes, with emissions from Canadian wildfires alone constituting a substantial portion of this total (Weisse et al. [2022]). As wildfires emit carbon dioxide, particulate matter, and various hazardous gases like carbon monoxide (CO), nitrogen oxides (NOx), and non-methane organic compounds (NMOC), they heighten the greenhouse effect, contributing to a warming planet (Naehler et al. [2007]). This release of carbon, especially in such large amounts, creates the feedback loop between wildfires and climate change. Increasing temperatures and drier conditions fuel the likelihood and intensity of future fires, thus creating a cycle of destruction and environmental degradation. These emissions also pose immediate health risks, causing respiratory diseases, aggravating pre-existing cardiovascular conditions, and have long-term implications ([Reid et al., 2016, Fadadu et al., 2021, Noah et al., 2023, Mangual et al., 2024]). Wildfires also impact the interconnectedness of air quality, ecosystem health, and biodiversity. These events fundamentally change habitats and ecosystems, affecting species composition, abundance, and resilience, thus playing a pivotal role in shaping biodiversity in both immediate and long-term contexts (Reisen et al. [2015], Jaffe et al. [2020], Burkle et al. [2015], Heil and Burkle [2018], Gates et al. [2021]). Due to these effects, it is crucial to delve into the roots of these destructive events. The leading causes of wildfires range from natural phenomena, such as lightning strikes, to human activities, including land clearing, campfires, and the improper disposal of cigarettes and faulty electric poles (Halofsky et al. [2020]). For instance, between 2013 and 2020, Turkey witnessed 950 wildfires caused by lightning strikes (Sari [2023]). Similarly, from 1990 to 2016, lightning was responsible for approximately 47 percent of all wildfire incidents in

Canada, highlighting its significant role in ignition (Tymstra et al. [2020]). These stress the crucial impact of natural causes, particularly lightning, on wildfire genesis. They reveal the complexity and diversity of ignition sources across different landscapes and highlight regional vulnerabilities influenced by topography and forest cover characteristics. Climate change emerges as a critical factor, amplifying conditions suitable for wildfires and influencing the ignitable materials available. Alterations in climate directly impact fuel availability and moisture content; for instance, during drought periods, potential fuels' increased abundance and dryness significantly elevate fire risk and intensity (Halofsky et al. [2020]). Understanding the variables that escalate wildfire severity is essential. Environmental factors such as drought conditions, elevated temperatures, strong winds, and low atmospheric humidity are critical in determining the likelihood and intensity of wildfire events. For example, numerous studies have identified temperature as a critical factor influencing annual wildland fire activity, with warmer conditions correlating with increased fire occurrences and severity (Coogan et al. [2021], Flannigan et al. [2005], Hély et al. [2001]). This relationship underlines the significance of temperature as a variable in fire dynamics, where higher temperatures can lead to drier conditions, enhancing fuel flammability and the potential for more extensive and intense wildfires (Wasserman and Mueller [2023], Alizadeh et al. [2021]). Wildfire behavior is influenced by a complex interplay of numerous environmental, meteorological, and vegetation-related factors. This complexity presents challenges in identifying the most critical features and accurately predicting wildfire occurrences. Given the escalating frequency of wildfires and their devastating effects, developing reliable prediction models has become increasingly important. These models are vital for mitigating risks, reducing damage, improving emergency response plans, and optimizing the allocation of firefighting resources. Machine-learning approaches have emerged as powerful tools for constructing predictive models for wildfires

in this era of data abundance. However, the complicated relationships among features in wildfire datasets often necessitate advanced techniques to handle their complexity. Metaheuristic algorithms address this, as they excel in exploring large search spaces and efficiently identifying optimal feature sets. These algorithms enable the development of more accurate models by selecting the most relevant features, thereby enhancing the predictive capabilities of machine-learning methods in wildfire studies. In this research, we evaluate a range of metaheuristic algorithms, including Atom Search Optimization (ASO), Barnacles Mating Optimizer (BMO), Chef Base Optimization (CBO), Energy Valley Optimizer (EVO), Equilibrium Optimizer (EO), Exponential Distribution Optimizer (EDO), Genetic Algorithm (GA), Golden Ratio Method (GRM), Liver Cancer Algorithm (LCA), Manta Ray Foraging Optimization (MRFO), Particle Swarm Optimization (PSO), Slime Mould Algorithm (SMA), and Walrus Optimization Algorithm (WOA). Building on the foundational understanding of the importance of predictive modelling in wildfire management and the innovative use of metaheuristic algorithms, this research sets out specific objectives to advance wildfire prediction. Our goals aim to improve the field of wildfire prediction and offer insights that can enhance risk management strategies. The primary aim is to leverage the strengths of selected metaheuristic algorithms to boost the accuracy, efficiency, and real-world applicability of wildfire prediction models. By doing so, we aim to contribute to more effective wildfire prevention and control efforts.

Research Objectives:

1. To apply and evaluate a diverse range of advanced metaheuristic algorithms, including Atom Search Optimization (ASO), Barnacles Mating Optimizer (BMO), Chef Base Optimization (CBO), Energy Valley Optimizer (EVO), Equilibrium Optimizer (EO), Exponential Distribution Optimizer (EDO),

Genetic Algorithm (GA), Golden Ratio Method (GRM), Liver Cancer Algorithm (LCA), Manta Ray Foraging Optimization (MRFO), Particle Swarm Optimization (PSO), Slime Mould Algorithm (SMA), and Walrus Optimization Algorithm (WOA), for feature selection in wildfire prediction models. This objective aims to determine which algorithms most effectively improve model performance in predicting wildfire occurrences.

2. To develop and refine the Spiral Liver Cancer Algorithm to enhance the standard LCA. It addresses limitations such as inefficient convergence and instability caused by giant Levy flight steps. By introducing Spiral updates, this objective seeks to enhance search precision and improve feature selection, outperforming other metaheuristic algorithms in wildfire prediction tasks.
3. To identify the most influential variables in wildfire prediction through the optimized feature selection processes provided by the metaheuristic algorithms. A clearer understanding of these critical variables will improve model accuracy and efficiency in predicting wildfire events.
4. To perform a comprehensive comparative analysis of the selected algorithms regarding prediction accuracy, computational efficiency, and scalability. This objective aims to pinpoint the most effective approaches for real-world applications in wildfire prediction, contributing to more informed risk management strategies.

Ultimate Goal: The primary goal of this research is to advance wildfire prediction by establishing effective feature selection techniques through a detailed evaluation of metaheuristic algorithms. By accomplishing this, the study aims to:

- Set a benchmark for the most efficient metaheuristic algorithms in wildfire prediction, enhancing the accuracy and robustness of predictive models.

- Improve the Spiral Liver Cancer Algorithm (LCA) to address the limitations of traditional approaches, ultimately outperforming other algorithms in precision and convergence.
- Identify the key variables most significantly influencing wildfire prediction, enabling more efficient and streamlined model development.
- Provide practical guidelines for selecting the most appropriate algorithms for real-world applications, supporting better risk management strategies and informed decision-making in wildfire prevention and control.

By focusing on these objectives, this research seeks to make a meaningful contribution to the growing wildfire prediction and management field. Through the insights gained, the study aims to enhance predictive accuracy and provide valuable information that supports strategic decision-making in wildfire mitigation efforts. The following chapters are designed to explore the research in a structured and detailed manner. Chapter 2 provides a thorough literature review, situating this study within the broader context of wildfire prediction research and identifying the existing gaps this work aims to address. Chapter 3 outlines the research design and methodology, particularly concerning the selection and application of metaheuristic algorithms for feature selection and developing and refining the Spiral Liver Cancer Algorithm (LCA). Chapter 4 presents the experimental results and offers a detailed analysis of the findings, discussing their practical applications and theoretical implications in the context of wildfire prediction. Finally, Chapter 5 concludes the thesis by reflecting on the research outcomes, their contributions to the field, and the opportunities for future research. This final chapter highlights how this study paves the way for further advancements in wildfire prediction strategies and risk management.

Chapter 2

Literature Review

Amid the escalating challenges posed by climate change, wildfires have become increasingly frequent and destructive, highlighting the critical need to enhance predictive capabilities. This literature review explores the evolving field of wildfire prediction and management, focusing on integrating advanced metaheuristic algorithms and machine learning to improve prediction accuracy and efficiency. As we delve deeper into the complexities of wildfire prediction, it becomes essential to fully leverage the power of machine learning—an area of artificial intelligence that enables systems to learn from data and make informed predictions. By combining machine learning models with metaheuristic algorithms, including Atom Search Optimization (ASO), Barnacles Mating Optimizer (BMO), Chef Base Optimization (CBO), Energy Valley Optimizer (EVO), Equilibrium Optimizer (EO), Exponential Distribution Optimizer (EDO), Genetic Algorithm (GA), Golden Ratio Method (GRM), Liver Cancer Algorithm (LCA), Manta Ray Foraging Optimization (MRFO), Particle Swarm Optimization (PSO), Slime Mould Algorithm (SMA), and Walrus Optimization Algorithm (WOA), represents a promising approach to

refining and improving the accuracy of wildfire prediction models.

The review is organized into three key sections. First, the "Research History of Wildfire Science" traces the evolution of wildfire research, providing the necessary context to appreciate the novel contributions of contemporary algorithms. Next, the "Application of Machine Learning in Wildfire Prediction" delves into how these advanced algorithms are applied to analyze large datasets, enabling the identification of patterns and predictors with greater precision. Finally, the "Optimizing Wildfire Prediction Models through Advanced Feature Selection" section highlights the importance of selecting relevant features for improving predictive accuracy and operational efficiency. This part critically examines the role of metaheuristic algorithms in refining feature selection, comparing their effectiveness and discussing their practical implications for wildfire management.

2.1 Research History of Wildfire Science In Canada

Before structured scientific studies on wildland fires began, Indigenous communities in Canada had developed advanced fire management practices, Such as controlled burns to manage ecosystems and protect communities, reflect a deep understanding of wildfire dynamics, nurtured over millennia of living in harmony with their environment (Hoffman et al. [2022], Huffman [2013], Christianson [2015]). This knowledge contrasts sharply with the fire suppression strategies introduced by European settlers. When European settlers arrived, their view of forest resources was mainly economic, with little attention to sustainable management or fire prevention. Early milestones in organized wildfire management include creating the

Timber, Mines, and Grazing Branch in 1882 and introducing forest rangers in 1883 (Johnstone [1991]). Quebec's 1883 initiative for forest rangers and Ontario's 1895 Forest Reserves Act further highlighted the growing need for structured wildfire prevention and management strategies (Rajala [2005]). In the 1920s, the formal study of wildfire science began to take shape, aided by the political climate under Mackenzie King's Liberal government. A national forest protection conference in 1924 marked a turning point, signalling the government's interest in fire control and hinting at a national forestry policy. However, progress toward a comprehensive policy stalled due to political concerns about federal-provincial relations. This resulted in the provinces taking control of natural resource management, leaving federal efforts, such as the Dominion Forest Service under Director E.H. Finlayson, in a precarious position. The Great Depression further strained resources, impacting both forestry research and management. Despite these setbacks, significant progress was made in wildfire science during this period, particularly at the Petawawa experiment station. Researchers James G. Wright and Herbert W. Beall studied how weather conditions impacted forest flammability, creating foundational knowledge to fuel further nationwide research efforts (Stocks2 et al. [1989], Paul [1969]). These studies led to the creation of the Canadian Forest Fire Weather Index (FWI) System in 1970 and the Forest Fire Behavior Prediction (FBP) System in the 1980s (Paul [1969], McALPINE et al. [1990], Wagner [1987]). Building on years of empirical research and technological advancements, these systems revolutionized wildfire prediction and management in Canada, leading to more precise and effective fire management strategies (Groot [1987], Stocks2 et al. [1989]). Another major development in wildfire prediction came in 1972 with the creation of Rothermel's Fire Spread Model. Although not developed in Canada, Rothermel's model globally impacted wildfire behaviour modelling, including in Canada (Andrews [2018]). This model introduced standardized fuel types and

accounted for wind and slope conditions, allowing for more accurate fire behaviour predictions. The Canadian FBP System integrated principles from Rothermel's work, adapting them to Canadian wildfires, and it remains a central tool in the country's fire management strategy. The development of wildfire prediction models in Canada did not stop there. In 1999, the Prometheus project aimed to create a more advanced simulation model for fire growth, building on the strengths of the FBP System. The goal was to integrate wave propagation theories, which allowed the model to simulate fire spread in more complex environments (Tymstra et al. [2010], Barber et al. [2009]). Prometheus marked a significant leap in wildfire modelling, combining detailed fire behaviour predictions with sophisticated mathematics, making it one of Canada's most important contributions to wildfire science. Despite the complexity of models like the FBP System and Prometheus, wildfires remain challenging to predict accurately, especially given Canada's evolving climate and various vegetation types. These models, while robust, face limitations in addressing the unpredictable variables introduced by climate change. A significant advancement in wildfire science has been the growing understanding of how weather affects wildfire behaviour. Studies have shown that temperature, lightning, wind, and humidity are critical in determining wildfire spread and intensity (Flannigan et al. [2005], Flannigan and Harrington [1988], Wotton [2009]). In particular, rising temperatures have been strongly linked to increased wildfire-affected areas (Flannigan et al. [2005], Jolly et al. [2015], BALSHI et al. [2009]). Additionally, higher temperatures contribute to frequent lightning strikes, a common cause of wildfires. Research suggests that a one-degree Celsius rise in temperature could result in a 5-100% increase in lightning activity, raising the risk of wildfires (Romps et al. [2014]). As our understanding of how weather influences wildfires deepens, the need for more dynamic fire models becomes clear. Traditional models, while effective, often struggle to integrate real-time weather data or adapt to rapidly

changing meteorological conditions. The field is shifting towards Next Generation Fire Modeling in response to these challenges. These advanced models aim to build upon frameworks like the FBP System by incorporating more complex variables, including refined climate change projections. Such innovations are designed to improve the accuracy of predictions and provide better tools for managing wildfires in an era of increasing unpredictability. The history of wildfire science in Canada shows a continuous evolution. From the early development of the FBP System to the incorporation of advanced meteorological data in Next-Generation Fire Modeling, each advancement has responded to emerging challenges in wildfire dynamics. Next-generation fire Modeling represents a critical advancement, integrating vast amounts of data and employing sophisticated algorithms to better predict and manage wildfires in a world where the risks continue to grow.

2.1.1 Related Work In Machine Learning and Wildfire Science

As the field of wildfire science progresses, it is crucial to note the emerging role of machine learning (ML) and artificial intelligence (AI) in reshaping our approach to understanding and managing wildfires. Integrating operations research, ML, AI, and other digital technologies marks a new era in wildfire science, offering unprecedented capabilities to analyze, predict, and respond to wildfire events. Machine learning, a subset of artificial intelligence, is defined as a set of methods that automatically detect patterns in data, facilitating the discovery of significant insights and enabling predictive models to anticipate future data trends or behaviours (Murphy [2012]). Machine learning's role within AI is akin to the brain's function within the human body, serving as a critical component that drives the intelligence and adaptability of AI systems. Three types of machine learning are pivotal

in analyzing and interpreting complex data sets: supervised, unsupervised, and reinforcement learning (Ivan et al. [2019]). Supervised learning involves training a model using a dataset where the desired output signals (labels) are already known, allowing the algorithm to learn by example (Kolosova and Berestizhevsky [2020], Bateman et al. [2020]). This methodology is particularly effective in wildfire science for predicting specific outcomes, such as fire spread or ignition likelihood, based on historical data where the results are already understood. Conversely, unsupervised learning does not rely on pre-labeled data. Instead, it attempts to identify patterns or intrinsic structures within the data itself (Geoffrey and Terrence J. [1999]). This approach can be instrumental in uncovering hidden patterns in wildfire occurrences or distributions, providing insights that might not be immediately evident through traditional analysis methods. Then, there is reinforcement learning, which stands out for its decision-making prowess. By interacting with an environment, the model learns to make a series of decisions, receiving feedback regarding rewards or penalties (Richard S. and Andrew G. [1998]). This dynamic learning process is akin to strategizing in complex and uncertain conditions, like developing adaptive fire suppression strategies in wildfire management, where the model iteratively improves its decision-making capability based on past outcomes. Having explored the core methodologies of machine learning—supervised, unsupervised, and reinforcement learning—we can now delve into how these techniques are applied to various facets of wildfire science, each playing a pivotal role in enhancing our understanding and management of wildfires. With their unique capabilities, these machine-learning approaches offer invaluable tools across different dimensions of wildfire research, from characterizing fire behaviour and assessing burn severity to improving fire detection and predicting occurrences. By applying these methods to specific wildfire-related challenges, researchers and practitioners can extract deeper insights, make more informed decisions, and develop more effective strategies for wildfire mitigation.

and management. Now, let us examine the nuanced applications of machine learning within critical areas of wildfire science, starting with the characterization of fire behaviour, moving through the estimation of burn areas and severity, advancing to the precision of fire detection, and understanding the patterns of fire occurrence.

Fire Fuel Characteristics

Understanding fire fuels is critical to accurately predicting wildfire behaviour (Nolan et al. [2016]). Fuel type, moisture content, density, and arrangement significantly influence how fires ignite, spread, and intensify. Tools such as the Fire Weather Index (FWI) have long provided crucial insights into fuel characteristics, supporting wildfire research and management (Heacutely et al. [2001], Emilio [2003], Wagner [1987]). These insights are vital for developing effective fire management strategies, refining predictive models, and ultimately minimizing the impact of wildfires. Recent advancements in remote sensing and machine learning have greatly improved our ability to assess fire fuels on a large scale. For example, Zhu et al. (2021) used a deep learning approach with a temporal convolutional neural network (TempCNN-LFMC) to estimate Live Fuel Moisture Content (LFMC) across the United States, achieving notable accuracy with an RMSE of 25.57 percent, which offers valuable data for real-time fire danger assessments (Zhu et al. [2021]). Similarly, Rivera et al. (2023) developed machine learning models to predict Critical Heat Flux for Ignition (CHFI) based on solid fuel properties, using over 2,000 experimental datasets and achieving high predictive accuracy (Rivera et al. [2023]). In the Mediterranean region, Cunill et al. (2022) applied a Random Forest algorithm to map LFMC, achieving solid results and demonstrating how these methods can improve fire danger assessments (Cunill Camprubí et al. [2022]). Similarly, Costa-Saura et al. (2021) used Sentinel-2 data combined with meteorological variables to model LFMC

variability across Spain, showing that seasonal changes in LFMC are crucial for predicting fire risk (Costa-Saura et al. [2021]). However, while moisture content is essential, it is only one part of the equation. Other critical factors—fuel type, density, and arrangement—also play a crucial role in determining wildfire behaviour. These attributes, shaped by the local vegetation and landscape, influence how fires ignite, spread, and increase in intensity. Fuel type significantly impacts wildfire behavior due to the varying flammability of different vegetation. Coniferous forests, for instance, are highly flammable because of their resinous composition, which encourages rapid fire spread. Hayes et al. (2024) found that in burned boreal forests, fuel constraints significantly limit fire behavior, making coniferous species particularly vulnerable to repeated, intense fires due to their flammability and the fuel they provide over time (Hayes et al. [2024]). Likewise, Abdollahi and Yebra (2023) noted that shrublands are composed of refined, dry fuels that ignite easily, especially under dry conditions, leading to faster and more intense fire spread (Abdollahi and Yebra [2023]). The density of vegetation also plays a significant role in fire intensity. Denser forests, such as those with high biomass density, tend to support more severe fires. For example, Boyd et al. (2023) observed that dense boreal forest ecosystems in Alaska, with large amounts of combustible material, including canopy and ground fuels, result in higher fireline intensity and faster surface fire spread (Boyd et al. [2023]). Albini (1976) also found that dense fuel loads contribute to longer flame lengths and higher heat outputs, which in turn increases fire severity (Station [1976]). On the other hand, Rothermel (1972) noted that forests with lower fuel density, such as open woodlands, tend to experience less intense fires due to reduced fuel loads, leading to quicker burn-out times and lower flame heights (Station [1972]). This shows that fuel type and the amount of available fuel are crucial in determining how wildfires behave. Finally, the arrangement of fuels—both vertical and horizontal—within an ecosystem is critical

in determining how fires move through the landscape. Forests with continuous vertical structures, from ground vegetation to the canopy, create a pathway for fires to transition from surface fires to more dangerous crown fires. Myroniuk et al. (2023) found that vegetation's vertical and horizontal continuity, particularly in the Chornobyl Exclusion Zone, allows for faster fire spread. This highlights how the spatial arrangement of fuels and their type and density can significantly influence the intensity and unpredictability of wildfire spread (Myroniuk et al. [2023]).

Burn Areas and Fire Severity

Wildfire research has advanced significantly with new methods for analyzing and understanding the severity of burned landscapes. In recent years, satellite imagery and machine learning have emerged as essential tools for quantifying the impact of wildfires and improving the accuracy of fire severity assessments. Satellites like Landsat-8, Landsat-9, and Sentinel-2 have become central to wildfire monitoring. These platforms capture high-resolution, multispectral data that allows researchers to differentiate between burned and unburned areas and categorize fire severity (McCorkel et al. [2018], Sigurdsson et al. [2022], Claverie et al. [2018], nas [2023a]). This data is crucial for assessing large, remote regions affected by fires, providing a comprehensive view of the aftermath. Machine learning has taken wildfire analysis to new heights, making burn severity mapping more accurate and efficient. For instance, Collins et al. (2018) found that a Random Forest classifier achieved over 95 percent accuracy in identifying major burn areas and 74 percent for lower severity burns using Landsat imagery. Similarly, Pu et al. (2004) used logistic regression and neural network models, achieving a remarkable 97 percent accuracy in identifying burned regions. These models underscore the potential of machine learning to enhance fire severity mapping by reducing the limitations of traditional

manual approaches. Combining different types of data has further refined these assessments. In Indonesia, researchers integrated Sentinel-1 radar and Sentinel-2 optical imagery, improving classification accuracy for burned areas to 91.80 percent and 95.80 percent, depending on the model used (Arjasakusuma et al. [2022]). This shows the power of using diverse data sources to create a more detailed understanding of wildfire impacts. Cloud-based platforms like Google Earth Engine are also gaining traction in fire mapping. Nelson et al. (2024) highlighted how these platforms could handle large-scale data processing, which is critical to tracking burn severity across vast landscapes and improving real-time fire management strategies (Nelson et al. [2024]). This capability is critical as we seek to automate and streamline wildfire monitoring efforts. Further work by Khankeshizadeh et al. (2024) combined Sentinel-1 radar with Sentinel-2 optical data, using deep learning techniques to achieve high accuracy in mapping forest burns (Khankeshizadeh et al. [2024]). The ability of machine learning to blend different types of data makes it an invaluable tool in the evolving landscape of wildfire research. Other research has shown that machine-learning techniques are adaptable to various ecosystems. Mitrakis et al. (2012) used a self-organizing neuro-fuzzy classifier to achieve over 95 percent accuracy with LANDSAT-5 TM imagery, outperforming traditional methods like neural networks and support vector machines. Tonbul et al. (2022) also employed ensemble learning algorithms like Random Forest and Canonical Correlation Forest (CCF), achieving robust results in both pixel-based and object-based burn severity classification (Tonbul et al. [2022]). These studies illustrate the versatility of machine learning for mapping fire-affected regions, regardless of ecosystem type. Recent advancements in deep learning have taken these techniques even further. Hu et al. (2023) used UNet and Attention UNet models to successfully segment burned areas in Landsat data, with a mean Intersection over Union (mIoU) of 0.78 and a Kappa coefficient approaching 0.90 (Hu et al. [2023]). In Alaska, deep

neural networks applied to MODIS data overcame class imbalance issues, achieving a recall score of 0.96 for identifying wildfire pixels (Langford et al. [2018]). One notable deep learning development is Seydi et al. (2022) Burnt-Net model, which uses Sentinel-2 data and deep learning morphological networks to map burned areas with an impressive accuracy of over 97 percent (Seydi et al. [2022]). This technology shows how post-fire assessments are becoming increasingly automated and precise. These developments collectively highlight a transformative phase in burn severity mapping, where machine learning and deep learning provide more nuanced insights and operational efficiencies. Machine learning and deep learning are transforming how we assess and respond to wildfires, offering more nuanced insights and improving analysis efficiency. As these techniques evolve, they can revolutionize fire severity mapping, real-time monitoring, and long-term wildfire management strategies. This integration of advanced technology is crucial for better preparing for the increasing wildfire risks posed by climate change.

Fire Occurrence and Detection

Accurate detection and monitoring of fire occurrences are essential for wildfire management and environmental conservation (Thapa et al. [2021]). Advances in remote sensing technology and computational methods have greatly enhanced our ability to detect and predict fires with increasing speed and accuracy. Historically, fire detection relied primarily on ground-based observations and reports. While valuable, these methods were limited in scope and often delayed. Satellite technology has revolutionized the field by providing real-time data over large and remote areas (Barmountis et al. [2020]). Satellites equipped with thermal sensors, such as MODIS (Moderate Resolution Imaging Spectroradiometer) and VIIRS (Visible et al.), have become essential for identifying heat signatures from active fires, offering

critical information for timely responses (nas [2023b]). Before the widespread use of satellite technologies, early mathematical models aimed at predicting fire occurrences were developed. For example, (Burrows et al. [1995]) applied a negative binomial model related to fire danger ratings, while (Cunningham and Martell [1973]) utilized a Poisson model based on fuel moisture levels. These early statistical models laid the groundwork for more advanced methods in fire dynamics research (Taylor et al. [2013]). Machine learning (ML) methods have emerged as prominent tools in predicting fire occurrences and detecting events in recent years. Standard ML techniques include Random Forests (RF), Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), and deep learning models such as Convolutional Neural Networks (CNNs). These advanced computational techniques offer greater accuracy and efficiency in analyzing complex datasets for fire detection and prediction, significantly advancing the field from earlier approaches. One early application of artificial neural networks (ANN) in wildfire science came in 1996 when (Vega-Garcia [1996]) applied ANN to predict human-caused wildfires in Alberta, Canada. Their model achieved an accuracy rate of 85 percent in predicting areas without fire and 78 percent in identifying areas with fire. Shortly afterward, in 2002, (Alonso-Betanzos et al. [2003]) used neural networks to estimate fire risk levels in Galicia, Spain, a wildfire-prone region. Their feedforward neural network model, with a 6-9-1 topology, performed well in classifying fire risks into four categories, with an accuracy of 77.2 percent. Similarly, the Fire Ignition Index (FII) was developed using neural networks, synthesizing indices such as the Fire Weather Index (FWI), Fire Hazard Index (FHI), and Fire Risk Index (FRI). The FII model showed high correct classification rates for fire instances (Vasilakos et al. [2007]). (Dutta et al. [2013]) analyzed ten neural network models to predict fire incidences based on climate data. The study found that the Elman Neural Network was the most effective, achieving an average accuracy of 94 percent, with other

models like the Time Delay Neural Network and Recurrent Neural Network also showing strong performance with accuracies of 88.21 percent and 92.77 percent, respectively. While ANNs are widely used in fire prediction, other models have also proven effective. For instance, SVMs have been applied to deduce fire hazard levels from meteorological data, demonstrating the versatility of machine learning methods (Sakr et al. [2010]). In one study, (Sakr et al. [2011]) compared ANN and SVM methods for forest fire prediction using minimal weather parameters such as relative humidity and cumulative precipitation, showcasing the potential to reduce costs and complexity. Additionally, (Yu et al. [2005]) developed a real-time forest fire detection and prediction system using a neural network within a wireless sensor network, emphasizing the importance of quick and accurate responses to emerging fire threats. In recent years, advanced neural network architectures, including CNNs and multilayer perceptrons (MLPs), have been widely adopted for global wildfire susceptibility prediction. Studies such as (Zhang et al. [2021]) offer insights into their comparative effectiveness and explore the interpretability of CNNs in this context. Random Forest (RF) use in fire occurrence prediction has gained momentum since 2012. Stojanova et al. [2012] conducted an extensive analysis comparing multiple machine learning techniques for predicting wildfire incidents, using a dataset that combined geographical, remote sensing, and meteorological data from Slovenia. Their study compared several classifiers, including K-Nearest Neighbors (KNN), Naïve Bayes, Decision Trees (J48 and jRIP), Logistic Regression (LR), SVM, and Bayesian Networks (BN). Another study by (Oliveira et al. [2012]) compared Multiple Linear Regression and RF methods to assess fire occurrence predictors in Europe, finding that RF provided superior predictive accuracy. The study also identified vital variables such as precipitation, soil moisture, and socio-economic factors as crucial for determining fire risk.

Feature Selection

In machine learning and predictive modelling, feature selection is a crucial process that significantly influences model performance, interpretability, and generalization. This process involves identifying and selecting a subset of relevant features (variables or predictors) for model construction, simplifying the models, reducing overfitting, and improving model robustness.

Feature selection is crucial in machine learning and predictive modelling, especially in complex domains like wildfire prediction. The process involves selecting a subset of relevant variables to improve prediction models' accuracy, efficiency, and interpretability. This step is crucial in wildfire prediction due to the high dimensionality of environmental, meteorological, and topographical data. While feature selection research in wildfire prediction is somewhat limited, it plays a pivotal role in identifying which environmental, meteorological, and topographical variables most significantly impact fire incidents and their spread. For example, Sadrabadi and Innocente [2023] demonstrated the application of feature selection techniques across various machine learning models such as Random Forest (RF), Extra Trees (XT), XGBoost (XGB), and CatBoost (CatB) in the context of wildfire prediction. The analysis revealed that certain features, including terrain slope and specific humidity indices, were consistently deemed less necessary across all models, suggesting their limited influence on fire prediction accuracy. Furthermore, Fernández-García et al. [2022] underscores the Normalized Difference Water Index (NDWI) as a pivotal predictor for assessing burn severity. Notably, NDWI, when applied with a cubic predictive model, demonstrated the most substantial correlation with burn severity among the evaluated variables. This relationship is characterized by an inverse association at intermediate NDWI values, as evidenced by the highest nRMSE and R^2 metrics, pinpointing NDWI's critical role in burn severity modelling. Building

upon that Wang et al. [2024] study, trained their model on sixteen features derived from detailed meteorological, human activity, topography, fuel characteristics, and geographical data, which was consistently selected across models for their predictive relevance. These features include average daily mean, maximum, and dew point temperatures (Tmean et al., respectively), maximum vapour pressure deficit (VPDmax), building coverage (BldgCover), slope, digital elevation model (DEM), and the Normalized Difference Vegetation Index (NDVI). Furthermore, Tracy et al. [2018] novel approach in ecological niche modelling for wildfire prediction leverages a unique random subset feature selection algorithm (RSFSA) to optimize the variable selection. Wang et al. (2024) took a similar approach by focusing on 16 features derived from meteorological data, human activity, topography, fuel characteristics, and geographical information. Their model emphasized the predictive power of variables like average daily mean temperature (Tmean), maximum vapour pressure deficit (VPDmax), building coverage (BldgCover), and Normalized Difference Vegetation Index (NDVI), demonstrating the importance of both human and environmental factors in predicting wildfire behaviour (Khosravi et al. [2023]).

Metaheuristic Algorithms for Enhancing Predictive Accuracy in Wildfire Science

Metaheuristic algorithms have significantly advanced the accuracy of wildfire prediction models by optimizing critical parameters that capture complex fire behaviours. These algorithms are integral to modern fire management strategies, offering rapid, reliable predictions for timely interventions. A landmark study by Pereira et al. (2024) demonstrated the potential of Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE) to refine the Rothermel model, a well-established wildfire spread predictor. This study optimized fuel

moisture, vegetation density, and wind speed. Among these, DE proved remarkably effective, achieving high precision with minimal adjustments, a crucial advantage in dynamic wildfire scenarios (Pereira et al. [2024]). Expanding on these findings, Jaafari et al. (2019) explored the integration of metaheuristics with fuzzy logic to enhance wildfire predictions further. They developed a hybrid model using the Adaptive Neuro-Fuzzy Inference System (ANFIS) combined with GA and the Firefly Algorithm (FA). The GA-ANFIS model, achieving an Area Under the Curve (AUC) of 0.92, illustrated the enhanced capability of this approach to model complex environmental interactions affecting wildfire risks (Jaafari et al. [2019]). In 2022, Pereira and colleagues revisited these techniques to concentrate on reducing prediction errors. Once again, DE stood out for its efficiency, adjusting the Rothermel model to deliver optimal predictions rapidly, aiding fire managers in resource allocation (Pereira et al. [2022]). Recent studies have shifted focus towards identifying the most influential factors for wildfire risk. In 2023, Zhang et al. developed a GA-optimized neural network that pinpointed critical predictors such as Historical Fire Density (HFD), Vegetation Type (VT), and Average Annual Temperature (AAT). This model's accuracy reached 83.7%, underscoring the importance of targeted feature selection in enhancing predictive precision (Zhang et al. [2023]). Advanced metaheuristic techniques were also applied to optimize neural networks and hybrid models. Nur et al. (2022) utilized Grey Wolf Optimizer (GWO) and Imperialist Competitive Algorithm (ICA) to optimize a Convolutional Neural Network (CNN) for mapping wildfire susceptibility, achieving an AUC of 0.974 and RMSE of 0.334 Nur et al. [2022]. Similarly, AlFugara et al. (2021) combined Support Vector Regression (SVR) and Adaptive Neuro-Fuzzy Inference System (ANFIS) with Whale Optimization Algorithm (WOA) and Simulated Annealing (SA), achieving AUROC values up to 0.965 Al-Fugara et al. [2021]. Azizi et al. (2023) also leveraged metaheuristics for hybrid feature selection in Iran,

identifying key predictors for enhancing regional wildfire risk mapping Azizi et al. [2023]. Adapting metaheuristics for specific regional conditions, Dieu Tien and colleagues tailored a Particle Swarm Optimization Neural Fuzzy (PSO-NF) model for Vietnam's unique environmental settings, such as high rainfall and dense vegetation. Their model achieved an AUC of 0.932, highlighting the adaptability of PSO to diverse ecological landscapes (Bui et al. [2017]). In a recent innovation, Tran et al. (2024) utilized emerging algorithms like Whale Optimization, Black Widow Optimization, and Butterfly Optimization in conjunction with XGBoost to predict wildfire risks in Hawaii. Their best model, BWO-XGBoost, achieved an AUC of 0.9269, showcasing the robustness of combining advanced algorithms with machine learning techniques to tackle the challenges of wildfire prediction across varied terrains (Tran et al. [2024]). Furthermore, Gharehchopogh et al. [2022] introduced the Random Subset Feature Selection Algorithm (RSFSA) as a novel approach for ecological niche modelling in wildfire prediction. This algorithm was particularly effective in identifying terrain slope and humidity indices as key predictors influencing wildfire occurrences. By systematically evaluating subsets of features, RSFSA enabled the refinement of predictive models by isolating variables with the highest impact on ecological niche dynamics. Metaheuristic algorithms add another layer to these models, improving prediction accuracy and optimizing model parameters. Algorithms like Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE) have become essential in fine-tuning models for wildfire prediction. Hybrid models, such as ANFIS-GA, PSO-NF, and GA-BPNN, which combine metaheuristics with fuzzy logic or machine learning, have shown great promise in the real-time mapping of wildfire susceptibility. These models help manage the complex environmental factors that drive wildfire behaviour despite the advancements and diverse applications of metaheuristic algorithms in wildfire science. Furthermore, implementing spiral dynamics in metaheuristic algorithms

for feature selection in wildfire science has been limited. Spiral optimization methods, known for their ability to escape local optima and explore the solution space more thoroughly, could significantly improve modelling the complex interactions of variables that influence wildfire behaviour and risks. Although spirals have yet to be used with metaheuristics in wildfire science, their demonstrated success in other fields provides compelling reasons to consider their application. For instance, the Linear Adaptive Spiral Dynamics Algorithm (LASDA) dynamically adjusts spiral parameters to enhance convergence and fitness accuracy, as demonstrated in system modelling. By leveraging adaptive linear spirals, LASDA has shown its utility in identifying parameters in flexible systems Nasir et al. [2016]. Similarly, the Spiral Search Mechanism integrated into the Equilibrium Optimizer (EO) employs logarithmic spirals to improve the diversity and robustness of search processes. This approach has proven effective in path planning for mobile robots. It highlights its potential to balance exploration and exploitation Ding et al. [2023], and Spiral Dynamics Optimization (SDO) uses logarithmic spirals to address global optimization challenges by balancing diversification and intensification. Its applications in engineering optimization further showcase its adaptability to high-dimensional problems Omar et al. [2022]. Furthermore, the Hybrid Particle Swarm Optimization with Spiral-Shaped Mechanism (HPSO-SSM) tackles the "curse of dimensionality" by employing geometric spirals for optimizing feature subsets, significantly improving classification accuracy in high-dimensional datasets Xie et al. [2021]. Despite the advancements and diverse applications of metaheuristic algorithms in wildfire science, specific gaps present opportunities for future research. Spiral dynamics have not been integrated into metaheuristic algorithms for feature selection in wildfire science, leaving unexplored potential to enhance model performance and address optimization challenges. Moreover, the Liver Cancer Algorithm (LCA), a recently introduced metaheuristic, has not yet been combined with spiral dynamics

to solve optimization problems or perform feature selection. Such an integration could significantly enhance its ability to navigate large, complex solution spaces. Finally, no studies have applied metaheuristic algorithms as feature selection tools for wildfire prediction in Canadian provinces, where diverse environmental factors influence wildfire behaviours. Addressing these gaps could provide transformative advancements in wildfire modelling and prediction.

Chapter 3

Methodology

The methodology chapter details the extensive steps to develop, implement, and assess a predictive wildfire model, addressing the complexity and class imbalance commonly encountered in wildfire datasets. This research applies advanced meta-heuristic algorithms for feature selection across eight Canadian provinces, aiming to enhance predictive accuracy by identifying critical environmental variables that significantly impact wildfire risk. The chapter is structured to clarify each component of the research design, from data collection and study area specification to algorithmic selection and model evaluation. The methodology follows a structured framework, beginning with an overview of the study's objectives, hypotheses, and overall approach, setting the foundation for the research. Data collection and sources are then explored in detail, specifying the data types used—such as satellite observations, meteorological data, vegetation indices, and historical wildfire occurrences—and discussing preprocessing steps to ensure data quality and consistency. Following this, the study areas are introduced, highlighting the unique environmental and climate characteristics of the eight selected Canadian provinces to enhance

the robustness and relevance of the analysis. In the core sections of this chapter, we delve into the research design by describing the theory behind each metaheuristic algorithm and providing details on implementation and parameterization. We also comprehensively outline the predictive modelling process, covering model training, validation, and testing procedures. Finally, evaluation metrics provide a structured approach for comparing and analyzing each algorithm's effectiveness. This ensures that the results align with the study's objectives and contribute valuable insights into wildfire prediction across diverse environmental settings.

3.1 Study Framework

This study follows a quantitative, empirical framework to advance wildfire prediction across diverse Canadian provinces through an in-depth evaluation and refinement of metaheuristic algorithms for feature selection. By leveraging data-driven methodologies, the research framework is structured to test the predictive capabilities of various metaheuristic algorithms and enhance their computational efficiency and suitability for wildfire prediction tasks. Particular attention is given to refining the Spiral Liver Cancer Algorithm (LCA), with improvements designed to resolve stability and convergence issues, strengthening its role in predictive modelling.

In addition, the framework emphasizes identifying the most influential environmental variables that drive wildfire occurrences within each unique provincial context. This approach aims to tailor model insights to the distinct climatic and environmental conditions across provinces, providing a more nuanced and accurate predictive capability. Through systematic analysis, the study framework ensures rigorous assessment and comparison of the selected algorithms, contributing to

wildfire risk management and the broader field of environmental predictive modelling.

3.1.1 Objective Of The Study

The primary objectives of this research are as follows:

- **Enhance Wildfire Prediction Accuracy:** Develop and optimize wildfire prediction models by applying metaheuristic algorithms for effective feature selection across diverse Canadian provinces.
- **Evaluate Metaheuristic Algorithms:** Test and compare multiple algorithms to determine their impact on feature selection and effectiveness in enhancing wildfire prediction models.
- **Algorithm Refinement and Development:** Integrate spiral updates to improve the standard Liver Cancer Algorithm (LCA). This refinement addresses convergence and stability issues, enhancing LCA's efficiency and accuracy in feature selection for wildfire prediction.
- **Identification of Key Predictive Variables:** Pinpoint the environmental variables most strongly associated with wildfire occurrences, focusing on optimizing model inputs for more accurate predictions.
- **Comparative Analysis:** Conduct a comprehensive analysis of each algorithm's predictive accuracy, computational efficiency, and scalability to identify the most suitable approaches for real-world wildfire risk management applications.

3.1.2 Hypotheses and Research Questions

The following hypotheses and research questions guide this study:

- **Hypothesis 1:** Training wildfire prediction models with selected features derived from metaheuristic algorithms will outperform models trained using all features, leading to better model efficiency and prediction accuracy.
- **Hypothesis 2:** The refined Spiral Liver Cancer Algorithm will outperform other metaheuristic methods regarding wildfire prediction accuracy and stability.
- **Hypothesis 3:** Each province will have unique environmental features contributing more significantly to wildfire occurrences, reflecting region-specific conditions impacting fire risk.

These hypotheses guide our study with the following research questions:

- **Research Question 1:** How effectively do metaheuristic algorithms improve feature selection accuracy and computational efficiency in wildfire prediction models compared to using all available features?
- **Research Question 2:** Does the Spiral Liver Cancer Algorithm provide superior feature selection strength and prediction accuracy performance compared to other metaheuristic methods?
- **Research Question 3:** Which environmental features most predict wildfire risk in each Canadian province, and how do these critical predictors vary regionally?

3.2 Data Collection and Sources

This study combines data from two primary sources: FIRMS (Fire Information for Resource Management System) for fire occurrence data and ERA5 for atmospheric and environmental data. These sources provide comprehensive and reliable datasets for wildfire prediction and feature selection across diverse Canadian provinces.

3.2.1 Data Types and Origin

The datasets utilized in this study include:

1. FIRMS (Fire Information for Resource Management System):

FIRMS offers global fire occurrence data derived from NASA’s MODIS and VIIRS sensors, with near real-time data available approximately three hours after the satellite overpass. The FIRMS Fire Map provides an inter-active library of active fire detections, enabling historical and current data analysis. This study utilizes archived data from 2010 to 2023 (NASA [2024]).

2. ERA5 Reanalysis Dataset: ERA5, produced by ECMWF (European Centre for Medium-Range Weather Forecasts), is a comprehensive reanalysis dataset covering climate and weather data from 1940 onwards. Reanalysis combines model data with observations worldwide into a complete and uniform dataset. ERA5 provides hourly atmospheric, ocean-wave, and land-surface data and includes a 10-member ensemble uncertainty estimate at three-hour intervals. This dataset captures various environmental and climatic conditions, including temperature, wind, humidity, and precipitation, which are crucial for identifying conditions contributing to wildfire occurrences(ERA [2024], Bell et al. [2021]).

Table 3.1: Summary of Features from FIRMS and ERA5 Data Sources

Source	Feature	Data Type	Description
FIRMS	Latitude	Float	Latitude coordinate of fire occurrence
FIRMS	Longitude	Float	Longitude coordinate of fire occurrence
FIRMS	Brightness Temperature (TI4)	Continuous	Brightness temperature measured by MODIS or VIIRS, used for detecting high-temperature events
FIRMS	Brightness Temperature (TI5)	Continuous	Secondary brightness temperature measure, complementing TI4 in assessing fire intensity
FIRMS	Scan	Float	Width of the satellite scan line at the fire's location, affecting spatial resolution
FIRMS	Track	Float	Length of the satellite track at the fire's location
FIRMS	Acquisition Date	Date	Date of fire detection
FIRMS	Acquisition Time	Integer	Time of fire detection in HHMM format
FIRMS	Satellite	Categorical	Satellite identifier (e.g., MODIS, VIIRS)
FIRMS	Instrument	Categorical	Instrument used for detection (e.g., MODIS, VIIRS)

Source	Feature	Data Type	Description
FIRMS	Confidence	Categorical (Low, Medium, High)	Confidence level in fire detection
FIRMS	Version	Categorical	FIRMS dataset version identifier
FIRMS	Fire Radiative Power (FRP)	Continuous	Energy output of the fire, indicating fire intensity
FIRMS	Day/Night	Categorical (Day, Night)	Indicates whether the fire detection occurred during day or night
ERA5	10m U Component of Wind	Continuous	Eastward wind speed component at 10 meters above ground
ERA5	10m V Component of Wind	Continuous	Northward wind speed component at 10 meters above ground
ERA5	2m Dewpoint Temperature	Continuous	Dewpoint temperature at 2 meters, affecting fuel moisture
ERA5	2m Temperature	Continuous	Air temperature at 2 meters, critical for ignition conditions
ERA5	High Vegetation Cover	Continuous	Fraction of high vegetation cover, contributing to fuel availability
ERA5	Leaf Area Index (High Vegetation)	Continuous	Leaf area index for high vegetation, showing vegetation density
ERA5	Leaf Area Index (Low Vegetation)	Continuous	Leaf area index for low vegetation, relevant for fire-prone areas
ERA5	Low Vegetation Cover	Continuous	Fraction of low vegetation cover

Source	Feature	Data Type	Description
ERA5	Mean Sea Level Pressure	Continuous	Mean atmospheric pressure at sea level
ERA5	Mean Wave Direction	Continuous	Average wave direction, relevant for coastal regions
ERA5	Mean Wave Period	Continuous	Average wave period
ERA5	Sea Surface Temperature	Continuous	Temperature of the sea surface, impacting local climate
ERA5	Significant Wave Height	Continuous	Height of combined wind waves and swell
ERA5	Surface Pressure	Continuous	Atmospheric pressure at the surface
ERA5	Total Cloud Cover	Continuous	Fraction of cloud cover, influencing solar radiation
ERA5	Total Precipitation	Continuous	Total precipitation, affecting fuel moisture
ERA5	Type of High Vegetation	Categorical	Classification of high vegetation type
ERA5	Type of Low Vegetation	Categorical	Classification of low vegetation type
ERA5	Downward UV Radiation at Surface	Continuous	Amount of UV radiation reaching the surface
ERA5	Surface Latent Heat Flux	Continuous	Heat exchange process affecting temperature and humidity

3.2.2 Preprocessing Steps

A series of preprocessing steps was conducted to ensure data quality and consistency across datasets from FIRMS and ERA5. These steps standardized the data, handled missing values, and applied transformations to enhance model performance.

Merging

This study created a unified dataset by combining environmental data from ERA5 with fire occurrence data from FIRMS, ensuring the data was aligned geographically and temporally. To achieve this, we rounded latitude and longitude values in both datasets to one decimal place so that data points would match in location. Additionally, we separated each dataset's date and time information into distinct fields, then standardized FIRMS timestamps by rounding them to the nearest three-hour interval to match ERA5's time intervals. Using these adjusted fields, we merged the datasets to connect fire occurrences with relevant environmental conditions.

To enhance reliability, we applied confidence filtering to the FIRMS data. For MODIS, we retained only high-confidence fire detections, while for VIIRS, we included all detections marked as "High," "Medium," or "Low." This filtering emphasized the most reliable detections from each satellite source. Each detection represented a distinct area, with fire areas estimated at 1 km² for MODIS and approximately 0.14 km² for VIIRS, allowing us to calculate the total fire area and support spatial analysis.

For further refinement, we used the DBSCAN clustering algorithm to group nearby detections, assuming closely located detections were part of the same fire.

After clustering, duplicate entries within each cluster were removed, leaving one representative data point per fire event, minimizing redundancy and improving data accuracy.

To create a comprehensive dataset, we processed each environmental variable separately for each year from 2010 to 2022. In the final integration step, we combined all variables into a master dataset, setting up a unified source ready for feature selection and predictive modelling.

Feature Engineering

Adding derived features informed by domain-specific knowledge can significantly enhance model accuracy in wildfire prediction. These features capture the complex dynamics governing wildfire behaviour. Below are the derived features and the reasoning for their inclusion:

Wind Speed (ws) **Rationale:** Wind speed is essential in wildfire spread, as higher speeds enable ember transport and intensify fire spread Prapas et al. [2022].

Equation 1: Wind speed calculation based on vector components.

$$ws = \sqrt{u10^2 + v10^2} \quad (3.1)$$

Wind Direction (wd) **Rationale:** Wind direction influences the path of embers, which affects the fire's spread Prapas et al. [2022].

Equation 2: Calculation of wind direction in degrees.

$$wd = \left(\arctan \left(\frac{v10}{u10} \right) \times \frac{180}{\pi} \right) \mod 360 \quad (3.2)$$

Equation 3: Adjusted formula to align with 360° scale.

$$wd = (wd + 180) \mod 360 \quad (3.3)$$

Relative Humidity (RH) Rationale: Lower humidity increases the likelihood of ignition, making it a crucial factor in wildfire risk Taneja et al. [2021].

Equation 4: Relative humidity based on the August-Roche-Magnus approximation.

$$RH = 100 \times \frac{\exp \left(\frac{17.625 \times T_d}{243.04 + T_d} \right)}{\exp \left(\frac{17.625 \times T}{243.04 + T} \right)} \quad (3.4)$$

Datetime Features Temporal features, such as day of the week, week of the year, month, and season, capture patterns in wildfire occurrences, identifying seasonality and periodic trends.

Day of the Week (day_of_week) Rationale: Certain days may show different wildfire frequencies due to human activity or weather patterns, and this feature allows the model to learn these patterns.

Week of the Year (week) Rationale: Wildfire activity varies seasonally, and the week number helps capture this, especially during peak wildfire seasons.

Month (month) **Rationale:** The month indicates the time of year, enabling the model to detect monthly trends in wildfire occurrences, which are often higher in warmer months.

Season (season) **Rationale:** Wildfire risk increases in specific seasons, such as summer. Mapping each month into a season provides insight into broader temporal patterns.

Season Mapping:

$$\text{season} = \begin{cases} \text{Spring} & \text{if month} \in \{3, 4, 5\} \\ \text{Summer} & \text{if month} \in \{6, 7, 8\} \\ \text{Autumn} & \text{if month} \in \{9, 10, 11\} \\ \text{Winter} & \text{if month} \in \{12, 1, 2\} \end{cases}$$

Dew Point Temperature Ranking (dewpoint_rank) **Rationale:** Dew point temperature affects wildfire risk by influencing atmospheric moisture. Lower dew points generally indicate drier conditions conducive to wildfire. The dew point ranking provides a straightforward feature representing wildfire risk due to atmospheric humidity.

Temperature Category (temperature_category) **Rationale:** Temperature is crucial in wildfire risk, as higher temperatures can increase likelihood and intensity. Categorizing temperature simplifies the model's interpretation of temperature's role in wildfire risk.

Categorization Logic:

- *Low*: Temperatures below 15°C
- *Medium*: Temperatures between 15°C and 25°C
- *High*: Temperatures above 25°C

Detailed Wildfire Risk Score (wildfire_risk) **Rationale:** The wildfire risk score is a calculated target variable synthesizing various environmental, seasonal, and atmospheric factors. This score captures conditions influencing fire occurrence, categorizing wildfire risk for training.

Supporting Factors in Calculation: The risk score integrates:

- **Precipitation:** Dry conditions increase fire risk, while precipitation reduces it (Wasserman and Mueller [2023], Abatzoglou and Kolden [2013], Abatzoglou and Williams [2016], Abatzoglou et al. [2017]).
- **Wind Speed:** Higher speeds help fires spread by supplying oxygen and dispersing embers (Liu et al. [2021]).
- **Vegetation Density:** Dense vegetation offers more fuel, raising the likelihood and intensity of wildfires (Loudermilk et al. [2022]).
- **Seasonal Influence:** Fire activity is higher in summer due to temperature and humidity patterns (Wasserman and Mueller [2023], Abatzoglou and Kolden [2013], Abatzoglou and Williams [2016]).
- **Cloud Cover:** Low cloud cover raises fire risk by allowing more sunlight and heat, while high cover reduces it (Drobyshev et al. [2021]).

- **Temperature and Humidity Interactions:** High temperatures, low humidity, and low pressure create favourable conditions for fires (Afolayan et al. [2024]).
- **Cluster Density Risk Score:** Historical fire clusters, identified through DBSCAN, adjust the risk score based on previous activity in the area.
- **Wind Direction:** South and Southeast winds, associated with drier air, increase fire risk (Fazel-Rastgar and Sivakumar [2022]).

This scoring approach aligns with research on environmental factors that influence wildfires. The calculated score, categorized into risk levels (Low, Moderate, High), gives the model essential insights for predicting wildfire risk.

Threshold-Based Risk Categorization:

$$\text{wildfire_risk} = \begin{cases} \text{High} & \text{if risk score} \geq 19 \\ \text{Moderate} & \text{if risk score} \geq 15 \text{ and } < 19 \\ \text{Low} & \text{if risk score} < 15 \end{cases}$$

The predictive model can better understand wildfire behaviour by incorporating thoughtfully designed features rooted in environmental science. Variables such as wind speed, relative humidity, and seasonality are essential in assessing wildfire risk. This process transforms complex data into clear and meaningful factors, improving the model's ability to make accurate and reliable predictions about wildfire risk.

3.3 Study Areas

This research examines datasets from eight Canadian provinces: Alberta, British Columbia, Manitoba, Ontario, Northwest Territories, Quebec, Saskatchewan, and Yukon. By including such a diverse range of regions, we aim to address the variability in environmental and climatic factors that influence wildfires, ensuring our model is comprehensive and adaptable.

Alberta

Alberta, located at 53.9333°N and 116.5765°W, spans approximately 661,848 km². The province's landscape transitions dramatically, from the rugged Rockies in the west to sprawling prairies in the east. Precipitation varies significantly, from a dry 300 mm in the southeast to over 600 mm in the mountainous regions. Wildfires are most common between May and September, fueled by hot, dry conditions and gusty winds. Alberta's varied geography makes it a fascinating case for studying how local terrain and weather impact wildfire behaviour.

British Columbia

British Columbia, at 53.7267°N and 127.6476°W, covers roughly 944,735 km² and is one of Canada's most ecologically diverse provinces. BC's landscape is as varied as its wildfire patterns, from lush coastal rainforests to the dry interior grasslands. The province's wildfire season runs from spring to early fall, with the highest risks in its arid interior during the summer. BC's unique interplay between elevation, temperature, and humidity offers critical insights into wildfire ignition and spread

dynamics.

Manitoba

Manitoba, situated at 53.7609°N and 98.8139°W, spans 647,797 km² and is characterized by its continental climate, with cold winters and warm, humid summers. The southern parts of the province receive more rainfall than the northern boreal forests, creating a gradient in fire risk. Wildfires typically peak from May to September, with weather conditions like dry spells and heatwaves significantly increasing the likelihood of fires. Manitoba's mix of forests and prairies makes it a challenging yet rewarding region for wildfire analysis.

Ontario

Ontario, located at 51.2538°N and 85.3232°W, is Canada's most populous province, covering about 1,076,395 km². Its climate ranges from the humid south to the subarctic north, with precipitation levels between 700 mm and over 1,000 mm annually. Wildfires are most active from April to October, with the northern and central regions facing the highest risks due to dense forests and periodic dry conditions. Ontario's landscape's sheer size and variability make it a key area for understanding how climate and geography interact to influence wildfire patterns.

Northwest Territories

The Northwest Territories, situated at 64.8255°N and 124.8457°W, is vast, covering approximately 1,346,106 km². This northern region experiences long, frigid winters and brief, mild summers. Annual precipitation is sparse, usually below 400 mm,

with most falling during summer. The wildfire season here peaks from June to August, driven by dry weather and strong winds in its extensive boreal forests. This region's remote and rugged nature presents significant challenges for wildfire management and prediction, making it a unique study area.

Quebec

Quebec, located at 52.9399°N and 71.3100°W, is Canada's largest province by area, spanning 1,542,056 km². Its climate shifts from the humid south, where cities like Montreal and Quebec City are located, to the cold, subarctic north. Wildfire occurs between May and September, with the northern boreal forests particularly susceptible. Quebec's vast and varied terrain requires wildfire models to account for densely forested areas and less vegetated zones, offering valuable insights into regional fire dynamics.

Saskatchewan

Saskatchewan, centered at 52.9399°N and 106.4509°W, encompasses about 651,900 km². Known for its expansive prairies and northern boreal forests, the province experiences a continental climate with warm summers and cold winters. Annual precipitation ranges from about 300 mm in the drier southwest to 500 mm in the forested north—the wildfire season peaks from May to September. Saskatchewan's blend of grasslands and forests provides a unique perspective on how different ecosystems respond to fire risks.

Yukon

Yukon, located at 64.2823°N and 135.0000°W, spans approximately 482,443 km². Its dramatic mountain ranges and subarctic climate define the territory, with long, harsh winters and brief, warm summers. Precipitation is limited, usually below 400 mm annually, with most of it as summer rain—wildfire season peaks from June to August, with fire activity influenced by dry spells and fluctuating temperatures. Yukon’s remote and rugged environment makes wildfire prediction complex, highlighting the importance of localized data in understanding fire behaviour.

While this research focuses on eight provinces across Canada, certain regions were excluded due to limited wildfire data availability or lower wildfire incidence, which would contribute minimally to the study’s focus on high-risk areas. Concentrating on regions with diverse and complex wildfire patterns, this research provides insights relevant to the areas most impacted by wildfires.

3.4 Feature Selection and Metaheuristic Algorithm Implementation

In predictive modelling, selecting the most informative features is crucial, particularly in applications like wildfire prediction, where vast datasets include complex interactions between variables. The objective of feature selection in this study is to identify a subset of predictive features that improve model accuracy and interpretability while reducing computational overhead. Given the high-dimensional nature of wildfire datasets, effective feature selection can streamline model performance by reducing noise and focusing only on variables that directly impact

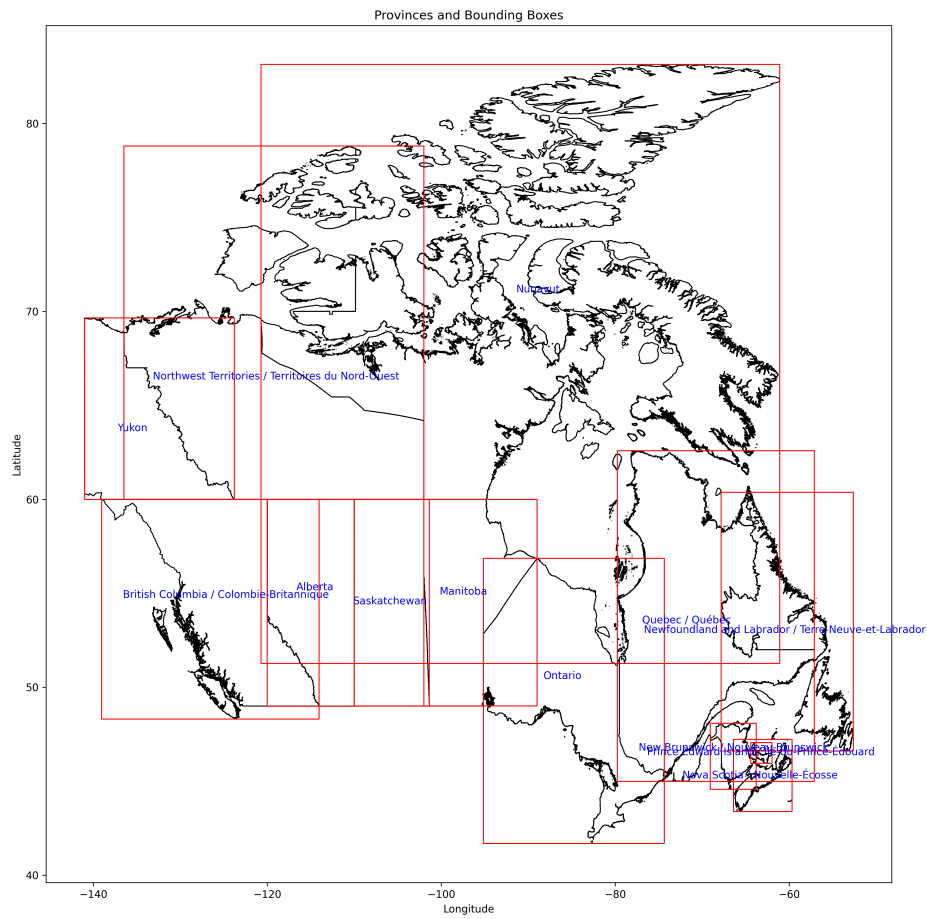


Figure 3.1: Map of Study Area across Canadian Provinces

wildfire risk.

3.4.1 Metaheuristic Algorithm Rationale

Metaheuristic algorithms are particularly advantageous in feature selection due to their adaptability in navigating large, complex search spaces and handling non-linear relationships among features. This study employs a variety of metaheuristic algorithms, each selected for its unique strengths in exploring and exploiting potential feature subsets.

This collection of algorithms includes population-based and solution-based methods, balancing exploration (diversifying the search) and exploitation (focusing on promising areas in the search space). Each algorithm contributes distinct search behaviours, enhancing the likelihood of identifying optimal or near-optimal subsets in the high-dimensional feature space of wildfire prediction datasets.

Wrapper Model for Feature Evaluation In this study, all metaheuristic algorithms employ a wrapper model approach, integrating a predictive model to evaluate the performance of each candidate feature subset (Kohavi and John [1997], Aboudi and Benhlina [2016]). The wrapper method is advantageous for feature selection as it iteratively assesses feature subsets based on predictive accuracy, allowing each algorithm to prioritize combinations that enhance model performance rather than relying solely on intrinsic characteristics of the data. A Random Forest (RF) classifier serves as the evaluation model here, with predictive quality measured through accuracy and F1-score.

In this wrapper framework:

- **Encoding Feature Subsets:** Each candidate solution, or agent, represents a unique subset of features. Using binary encoding, features are denoted as

included (1) or excluded (0), allowing metaheuristic algorithms to explore a range of feature combinations effectively.

- **Performance Evaluation:** For each feature subset, the wrapper model trains an RF classifier, assessing its predictive accuracy. This feedback guides the search process, allowing algorithms to adjust their exploration toward subsets that demonstrate improved model performance.

In the wrapper model’s evaluation process, we defined a fitness function for each agent (feature subset), using a cross-validated XGBoost classifier to measure its predictive effectiveness. This function calculates the fitness based on cross-validated F1-scores, providing critical feedback to the metaheuristic algorithms. The fitness function is represented as follows:

Equation 5: Fitness Function for Feature Subset Evaluation

$$\text{fitness}(A) = \begin{cases} 0 & \text{if } \sum A = 0 \\ \frac{1}{k} \sum_{i=1}^k \text{F1}(y_{\text{cv_val}}^{(i)}, \hat{y}_{\text{cv_val}}^{(i)}) & \text{otherwise} \end{cases} \quad (3.5)$$

where:

- A is the binary vector representing the feature subset the agent chooses.
- k is the number of folds in cross-validation (set to 10 in this study).
- $y_{\text{cv_val}}^{(i)}$ and $\hat{y}_{\text{cv_val}}^{(i)}$ are the true and predicted labels for the validation set in fold i , respectively.
- $\text{F1}(\cdot)$ denotes the F1 score function with a macro averaging strategy.

If no features are selected (i.e., $\sum A = 0$), the fitness score defaults to zero as a penalty. For agents selecting at least one feature, the mean F1 score across cross-validation folds provides a fitness measure, with a higher score indicating a more informative feature subset. The feature selection ratio is given by:

Equation 6: Feature Ratio Calculation

$$\text{feature ratio} = \frac{\sum A}{\text{len}(A)} \quad (3.6)$$

This represents the proportion of selected features relative to the total feature set, balancing feature inclusion with computational efficiency.

The metaheuristic algorithms iteratively refine feature subsets through this wrapper-based approach, maximizing model performance by selecting the most predictive features within wildfire datasets.

Key Parameters and Common Inputs/Outputs While each algorithm has unique characteristics, they share common parameters and outputs, such as:

- **Iterations (limit):** The number of cycles each algorithm undergoes influences its convergence behaviour.
- **Population Size:** The number of candidate solutions (or agents) explored in each iteration, balancing computational efficiency with the depth of exploration.
- **Training Data (X_{train} and y_{train}):** The input dataset comprising the features and corresponding target variable is used to evaluate feature subsets.

Common Outputs:

- **Best Fitness Value (best_fit_val):** The optimal fitness score achieved, reflecting the quality of the selected feature subset.
- **Fitness History (fitness_history):** Records fitness values over iterations, useful for analyzing the algorithm’s convergence behaviour.
- **Accuracy History (accuracy_history):** Tracks accuracy values over iterations, providing insight into predictive performance.
- **Feature Ratio History (feature_ratio_history):** Shows feature ratios across iterations, indicating the proportion of features selected.

3.4.2 Algorithm Descriptions

Atom Search Optimization (ASO)

Atom Search Optimization (ASO) is a physics-inspired metaheuristic algorithm designed to solve complex optimization problems by mimicking atomic motion in molecular dynamics. In ASO, each solution (or atom) interacts with others through forces derived from molecular physics, allowing for both exploration of the search space and intensification around promising solutions Zhao et al. [2018].

Algorithm Mechanics ASO is based on two main types of forces:

- **Interaction Force:** Derived from the Lennard-Jones potential, this force represents attraction and repulsion between atoms, enabling atoms to either converge around promising solutions or disperse to explore new areas.

- **Constraint Force:** A secondary force directing atoms toward the best-known solution, encouraging the algorithm to refine optimal solutions.

The position of each atom in ASO represents a potential solution, and its "mass" (related to the solution's fitness) influences its acceleration. The algorithm balances exploration and exploitation by adjusting the interaction forces over iterations, with lighter atoms exploring new regions and heavier atoms focusing on optimal solutions.

Parameter Settings The main parameters in ASO include:

- **Iterations:** Controls the algorithm's stopping condition based on convergence requirements.
- **Population Size:** The number of atoms, balancing computational efficiency with search robustness.
- **Depth Weight** and **Multiplier Weight:** Parameters that control the strength of interaction and constraint forces, influencing ASO's balance between exploration and exploitation.

Application to Feature Selection ASO's ability to avoid local optima and dynamically adapt to high-dimensional spaces makes it particularly useful for feature selection in wildfire prediction. By selecting a subset of relevant features, ASO enhances model performance while reducing computational costs, making it ideal for large datasets with complex interrelationships Zhao et al. [2019].

Algorithm 1 Inputs and Outputs

Inputs:

- X_{train} : Training data (features)
- y_{train} : Training data (target)
- atomno: Number of atoms (agents) to maintain diversity in the search space.
- dim: Dimension of the feature space (number of features) being evaluated.
- limit: Maximum number of iterations, controlling the depth of the search process.
- α, β, ω : ASO-specific hyperparameters influencing the balance between exploration and exploitation.

Outputs:

- best_atom: Best feature subset (binary vector) discovered by the algorithm.
- best_fit_val: Best fitness value achieved, representing the quality of the selected feature subset.
- fitness_history: History of fitness values over iterations, useful for analyzing convergence behaviour.
- accuracy_history: Accuracy values over iterations, providing insight into predictive performance.
- feature_ratio_history: Feature ratios over iterations indicate the proportion of selected features.

Algorithm 1 Atom Search Optimization (ASO) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each atom i in range $(0, \text{atomno})$ **do**


```

3:   Generate a random binary vector of length dim to represent selected features
4:   Initialize velocities for each feature in the atom
5: end for
6: Step 2: Calculate Initial Fitness
7: for each atom  $i$  do
8:   Perform wrapper model evaluation using cross-validation with XGBoost
      (see Equation 5).
9:   Store fitness, accuracy, and feature ratio for each atom
10: end for
11: Find and store the best initial solution (best_atom and best_fit_val)
12: Step 3: Optimization Loop
13: for iteration curr in range (1, limit + 1) do
14:   for each atom  $i$  do
15:     Update velocity using the formula:
        
$$\text{velocity}[i] = \text{velocity}[i] + \beta \times (\text{best\_atom} - \text{population}[i])$$

16:     Update position using V-function:
17:     for each feature  $j$  in atom  $i$  do
18:       Set position[ $j$ ] = 1 if  $V(\text{velocity}[i][j]) > \text{random}()$  else 0
19:     end for
20:     Recalculate fitness for the new feature subset
21:     Update best solution if the current fitness is better than best_fit_val
22:   end for
23:   Track fitness_history, accuracy_history, and feature_ratio_history
24: end for
25: Step 4: Return Results
26: Output:   best_atom,   best_fit_val,   fitness_history,   accuracy_history,
      feature_ratio_history

```

Barnacles Mating Optimization (BMO)

The Barnacles Mating Optimizer (BMO) is a bio-inspired metaheuristic algorithm that simulates the mating behaviours of barnacles. BMO emulates the biological processes observed in barnacle reproduction, where individuals selectively mate within a constrained range or disperse sperm broadly when mates are unavailable nearby (Sulaiman et al. [2020]).

BMO's optimization relies on two core dynamics:

- **Mating Selection and Penis Length (Interaction Constraints):** Each barnacle selects mates within a limited "penis length" (pl) distance, promoting exploitation by mating with close solutions to refine promising regions. If no suitable mate is within range, sperm casting occurs, simulating broad dispersal to enable distant interactions, enhancing exploration and avoiding local optima.
- **Inheritance of Parental Traits (Reproductive Process):** Offspring are generated by combining traits from selected parent barnacles based on the Hardy-Weinberg principle, which probabilistically blends characteristics. This balances exploration and exploitation, adapting offspring traits across iterations to refine potential solutions.

Parameter Settings in BMO: Fine-tuning these parameters is key to BMO's success:

Penis Length (pl): Controls mating range, balancing exploration and exploitation. Population Size: Sets the number of barnacles, impacting computational cost and diversity. Iterations: Determines the number of optimization cycles based on convergence needs.

Algorithm 2 Barnacles Mating Optimizer (BMO) for Feature Selection

```

1: Step 1: Initialize Population
2: for each barnacle  $i$  in range (0, pop_size) do
3:     Generate a random binary vector of length dim representing selected features
4: end for
5: Step 2: Calculate Initial Fitness
6: for each barnacle  $i$  do
7:     Perform wrapper model evaluation using cross-validation with XGBoost
        (see Equation 5).
8:     Store fitness for each barnacle
9: end for
10: Identify and store the best initial solution (best_barnacle and best_fitness)
11: Step 3: Optimization Loop
12: for each iteration in range (1, max_iter + 1) do
13:     for each barnacle  $i$  do
14:         Find local mates within mating_radius
15:         if no local mates or random()  $\geq$  sperm_cast_chance then
16:             Select a global mate randomly (sperm casting)
17:         else
18:             Select a random local mate
19:         end if
20:         Generate offspring by:
            a) Performing crossover between barnacle and mate

```

```

        b) Applying mutation based on the distance between barnacles
21:     Evaluate fitness of offspring
22:     if offspring fitness is better than current barnacle fitness then
23:         Replace barnacle with offspring
24:     end if
25: end for
26: Track best_fitness for each iteration in fitness_history
27: end for
28: Step 4: Return Results
29: Output: best_barnacle, best_fitness, fitness_history

```

Chef-Based Optimization Algorithm (CBOA)

The Chef-Based Optimization Algorithm (CBOA) is an optimization method inspired by the structured learning process in culinary arts. In CBOA, each candidate solution represents either a chef instructor or a cooking student, each playing a different role in reaching the best possible solution.

Algorithm Mechanics CBOA operates through two main phases:

- **Chef Instructors' Phase:** Chef instructors refine their skills by observing the techniques of the top chef (global search) and adding their improvements (local search). This phase helps balance the exploration of new solutions with the refinement of existing ones.
- **Cooking Students' Phase:** Students learn in three ways—by working with a chosen chef instructor, picking up a new skill from another chef, and

practicing independently. This diverse learning approach allows students to explore various solutions while focusing on improving specific areas.

In CBOA, the positions of chef instructors and cooking students represent potential solutions. Chef instructors guide the search toward promising areas while students focus on refining solutions nearby. Together, these roles help the algorithm balance exploring new options and intensifying efforts around the best solutions found so far. Through repeated training cycles, CBOA moves closer to finding the optimal solution (Trojovská and Dehghani [2022]).

Algorithm 3 Chef-Based Optimization Algorithm (CBOA) for Feature Selection

```
1: Step 1: Initialize Population
2: for each agent  $i$  in range (0, SearchAgents) do
3:     Generate a random binary vector of length dimension representing selected
       features
4: end for
5: Step 2: Calculate Initial Fitness
6: for each agent  $i$  do
7:     Perform wrapper model evaluation using cross-validation with XGBoost
       (see Equation 5).
8:     Store fitness for each agent
9: end for
10: Identify and store the best initial solution (top_chef and best_fitness)
11: Step 3: Optimization Loop
12: for each iteration in range (1, Max_iterations + 1) do
13:     Assign agents as "chef instructors" or "cooking students" based on fitness
       ranking
```

```

14:   for each agent  $i$  do
15:       if agent  $i$  is a chef instructor then
16:           if random()  $\leq$  0.5 then
17:               Follow top_chef to exploit best-known solution
18:           else
19:               Perform independent exploration with Levy flight
20:           end if
21:       else ▷ Agent  $i$  is a cooking student
22:           Select learning strategy randomly:
23:           a) Learn from a random chef instructor
24:           b) Master a specific skill from the top chef
25:           c) Self-improve through Levy flight for exploration
26:       end if
27:       Evaluate fitness of updated agent
28:       if agent's fitness is better than best_fitness then
29:           Update best_fitness and top_chef with current agent
30:       end if
31:   end for
32:   Track best_fitness for each iteration in fitness_history
33: end for
34: Step 4: Return Results
35: Output: top_chef, best_fitness, fitness_history

```

Equilibrium Optimizer (EO)

The Equilibrium Optimizer (EO) is a physics-inspired optimization method modelled after mass balance concepts in a control volume. In EO, each candidate

solution represents a particle with an evolving position based on concentration and balance principles, aiming to reach an equilibrium state.

Algorithm Mechanics EO operates through three main components:

- **Equilibrium Pool and Candidates:** EO maintains a pool of top solutions, representing equilibrium candidates. These candidates guide other particles by attracting them toward promising areas while retaining some variability to explore diverse regions.
- **Exponential Term (F):** This term is calculated based on the turnover rate (λ), facilitating a dynamic shift between exploration (wide-ranging searches) and exploitation (focused refinement) as iterations proceed. The term enables particles to expand their search early on and refine positions in later stages.
- **Generation Rate (G):** As a local search enhancer, this term directs particles to adjust their positions around candidates, particularly in smaller steps, reinforcing the search's precision as particles near equilibrium.

In EO, particles act as potential solutions, each adjusting its "concentration" based on local and global interactions. Through the equilibrium pool and the balance between F and G terms, EO effectively balances global exploration and local exploitation. Repeated interactions move EO closer to the global optimum solution (Faramarzi et al. [2020]).

Algorithm 4 Equilibrium Optimizer (EO) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each agent i in range (0, partCount) **do**

- 3: Generate a random binary vector of length dimension representing selected features
- 4: **end for**
- 5: Initialize the equilibrium pool eqPool with size poolSize, storing the best solutions found
- 6: **Step 2: Calculate Initial Fitness**
- 7: **for** each agent i **do**
- 8: Evaluate fitness using cross-validation with XGBoost (see Equation 5)
- 9: Store fitness for each agent in fitness_scores
- 10: **end for**
- 11: Identify the best initial solutions and update eqPool and eqfit (fitness of equilibrium candidates)
- 12: **Step 3: Optimization Loop**
- 13: **for** each iteration in range $(1, \text{max_iter} + 1)$ **do**
- 14: **for** each agent i **do**
- 15: Select an equilibrium candidate C_{eq} from eqPool
- 16: Generate a random vector λ to guide convergence and diversity
- 17: Calculate:
 - a) **Concentration Difference** term, FVec:

$$\text{FVec} = 2 \times \text{sign}(\text{rand} - 0.5) \times \left(\exp(-\lambda \times (1 - \frac{\text{iteration}}{\text{max_iter}})^{2 \cdot \text{iteration}/\text{max_iter}}) - 1 \right)$$
 - b) **Generation Rate** term, G:

$$\text{G} = 0.5 \times \text{rand} \times (C_{eq} - \lambda \times \text{agent}[i])$$
- 18: Update agent's position:

$$\text{agent}[i] = \text{clip}(\text{agent}[i] + \text{G} - \text{FVec} \times (\text{agent}[i] - C_{eq}), 0, 1)$$
- 19: **end for**

- 20: Evaluate fitness for each agent after update and update eqPool with the best solutions if necessary
 - 21: Track the best fitness values for each iteration in fitness_history
 - 22: **end for**
 - 23: **Step 4: Return Results**
 - 24: **Output:** The best agent, best fitness score, and fitness history over iterations
-

The Exponential Distribution Optimizer (EDO)

The Exponential Distribution Optimizer (EDO) is a mathematics-inspired algorithm rooted in the principles of the exponential probability distribution. The primary mechanism of EDO is based on exploiting the statistical properties of exponential distributions to drive both exploration and exploitation within an optimization landscape.

Algorithm Mechanics EDO comprises two main phases inspired by exponential distribution properties:

- **Exploitation Phase:** This phase is designed around the memoryless property of exponential distributions. EDO maintains a memoryless matrix where newly generated solutions are stored regardless of fitness. By disregarding past performance, EDO allows unsuccessful solutions (losers) to influence future generations. This phase also incorporates the guiding solution, a composite of the top-performing solutions, to enhance exploitation near promising areas.
- **Exploration Phase:** This phase uses the average solution (mean of all solutions) alongside randomly selected high-performing solutions to drive

exploration. By adjusting the contribution of each selected solution, EDO dynamically explores new areas and adjusts for diversity within the search space.

Through repeated cycles, EDO alternates between these phases, enabling a balance between exploring new areas and refining promising solutions. A switching parameter regulates this balance, ensuring the search process remains dynamic.

In EDO, guiding solutions, memoryless storage, and controlled exponential variance are pivotal, helping the algorithm move toward optimal areas and avoid getting trapped in local minima Abdel-Basset et al. [2023].

Algorithm 5 Exponential Distribution Optimizer (EDO) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each agent i in range $(0, N)$ **do**
- 3: Generate a random continuous vector within bounds $[LB, UB]$ for each dimension, representing selected features
- 4: **end for**
- 5: Set $BestSol \leftarrow 0$, $BestFitness \leftarrow -\infty$, and initialize solution matrix $Solutions$
- 6: **Step 2: Calculate Initial Fitness**
- 7: **for** each agent i **do**
- 8: Select features based on agent i 's continuous vector, converting values above a threshold to 1 (selected) and others to 0
- 9: Evaluate fitness using cross-validation with XGBoost on the selected features (refer to Equation 5)
- 10: Store fitness score in $Fitness[i]$
- 11: **end for**
- 12: Identify the best initial solution, updating $BestSol$ and $BestFitness$ with the top fitness and corresponding agent

13: **Step 3: Optimization Loop**

14: **for** each iteration in range (1, Max_iter + 1) **do**

15: **for** each agent i **do**

16: Generate a random vector λ to guide the update for each dimension

17: Select a random **guiding solution**, $C_{guiding}$, from the current population

18: Calculate:

 a) **Concentration Difference** term, FVec:

$$FVec = 2 \times \text{sign}(\text{rand} - 0.5) \times \left(\exp(-\lambda \times \left(1 - \frac{\text{iteration}}{\text{Max_iter}}\right)^{2 \cdot \text{iteration}/\text{Max_iter}}) - 1 \right)$$

 b) **Generation Rate** term, G:

$$G = 0.5 \times \text{rand} \times (C_{guiding} - \lambda \times \text{Solutions}[i])$$

19: Update agent i 's position:

$$\text{Solutions}[i] = \text{clip}(\text{Solutions}[i] + G - FVec \times (\text{Solutions}[i] - C_{guiding}), LB, UB)$$

20: **end for**

21: **Step 4: Evaluate and Update Fitness**

22: **for** each agent i **do**

23: Select features from the updated position of agent i (based on thresholding)

24: Recalculate fitness using cross-validation with XGBoost on the selected features

25: **if** new fitness of agent i is better than BestFitness **then**

26: Update BestFitness and BestSol with agent i 's solution and fitness

27: **end if**

28: **end for**

29: Record fitness values for each iteration in fitness_history

30: **end for**

31: **Step 5: Return Results**

32: **Output:** The best solution BestSol, best fitness score BestFitness, and statistics such as mean fitness, worst fitness, and feature selection ratio

Energy Valley Optimizer (EVO)

The Energy Valley Optimizer (EVO) is a metaheuristic optimization algorithm inspired by physical principles surrounding particle stability and decay processes. In EVO, candidate solutions act as particles, each possessing unique stability characteristics, guiding them toward more stable states within the solution space. This process involves different decay schemes—alpha, beta, and gamma decay—which mimic the real-life behaviour of particles moving toward stability (Azizi et al. [2023]).

Algorithm Mechanics EVO operates through three main phases, mirroring different particle decay modes:

- **Alpha Decay Phase:** In this phase, particles emit alpha particles, moving towards the global best solution by modifying specific attributes. This enables local search within a neighbourhood of promising solutions.
- **Beta Decay Phase:** Particles with high instability levels undergo beta decay, making larger jumps toward the stability band (optimal solution) by moving closer to both the best solution and a central point, increasing exploration.
- **Gamma Decay Phase:** Particles experiencing gamma decay adjust their positions based on neighbouring solutions, facilitating refined adjustments

for better stability within a localized region.

Algorithm 6 Energy Valley Optimizer (EVO) for Feature Selection

```
1: Step 1: Initialize Particles
2: for each particle  $i$  in range (1, Population Size) do
3:   Randomly generate initial positions within bounds for each particle, representing selected features
4: end for
5: Define Enrichment Bound (EB) based on initial stability levels of particles
6: Step 2: Calculate Initial Stability Levels
7: for each particle  $i$  do
8:   Select features based on the particle's position vector, converting values above a threshold to 1 (selected) and others to 0
9:   Evaluate fitness using cross-validation with XGBoost for each particle's selected features (see Equation 5)
10:  Assign initial stability level based on fitness score
11: end for
12: Determine best and worst stability levels as Best Stability (BS) and Worst Stability (WS)
13: Step 3: Optimization Loop
14: for each iteration in range (1, Max Iterations) do
15:   for each particle  $i$  do
16:     if Neutron Enrichment Level ( $NEL_i > EB$ ) then
17:       if Stability Level ( $SL_i > \text{Stability Bound}$ ) then
18:         Perform Alpha Decay:
19:         Adjust particle's position by moving it towards BS for local refinement
20:         Generate new candidate solution  $X_{New_i}$  by applying alpha decay
21:       else
```

```

21:         Perform Gamma Decay:
           Adjust based on neighbouring particles to refine the position
22:         Generate new candidate solution  $X_{New2_i}$  through gamma decay
23:     end if
24: else
25:     Perform Beta Decay for more unstable particles:
26:     Adjust particle  $i$  by moving it towards both BS and the central point
       of the population for wider exploration
27: end if
28: end for
29: Evaluate and update stability for each new candidate solution using
       XGBoost-based fitness evaluation
30: Update EB based on new stability levels and check stopping criteria
31: end for
32: Step 4: Return Results
33: Output: The particle with the highest stability (best fitness), fitness history
       across iterations, and selected features

```

Genetic Algorithm (GA)

The Genetic Algorithm (GA) is a classic metaheuristic optimization method inspired by principles of natural selection and genetics. In GA, candidate solutions, individuals, form a population that evolves over generations. A fitness function evaluates each individual, and the fittest individuals are more likely to pass their traits to the next generation, gradually steering the population toward optimal solutions (H. et al. [2009], Katoch et al. [2021]).

Algorithm Mechanics GA operates through three main evolutionary processes:

- **Selection:** Individuals are selected from the population based on their fitness scores. Higher fitness increases the likelihood of selection, favouring stronger solutions. Selection methods include roulette wheel selection, tournament selection, and rank-based selection.
- **Crossover (Recombination):** Selected individuals (parents) exchange parts of their genetic material to produce offspring. This process combines traits from both parents, allowing the population to explore new areas of the solution space while inheriting favourable characteristics. Common crossover methods include single-point, multi-point, and uniform crossover.
- **Mutation:** Offspring undergo mutation with a low probability, introducing small, random changes to their genetic material. Mutation maintains genetic diversity in the population and prevents premature convergence on local optima. It is typically implemented by flipping bits in a binary representation or making small numerical changes in real-valued encoding.

In GA, the evolving population undergoes these three processes in each generation. The algorithm continues until a termination condition is met, such as reaching a maximum number of generations or achieving a specified fitness level.

Algorithm 7 Genetic Algorithm (GA) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each individual i in range (1, Population Size) **do**
- 3: Generate a random binary vector of length num_features, representing selected features
- 4: **end for**

```

5: Step 2: Calculate Initial Fitness
6: for each individual  $i$  do
7:     Evaluate fitness using cross-validation with XGBoost
8:     Store fitness score for each individual
9: end for
10: Identify the best initial solution and store it as Best Solution with fitness
    Best Fitness
11: Step 3: Optimization Loop
12: for each generation in range (1, Max Generations) do
13:     Initialize a new population for the next generation
14:     for each pair of parents in population do
15:         Select two parents using tournament selection
16:         if random()  $\leq$  crossover rate then
17:             Perform single-point crossover on parents to create two children
18:         else
19:             Children are identical to parents (no crossover)
20:         end if
21:         Perform mutation on each child by flipping bits with probability equal
            to the mutation rate
22:         Add children to the new population
23:     end for
24:     Replace the current population with the new population
25: Step 4: Calculate Fitness for New Population
26: for each individual  $i$  in the new population do
27:     Evaluate fitness using cross-validation with XGBoost
28:     Store fitness score for each individual
29: end for

```



```

30:   Identify the best solution in the new generation
31:   if new best solution's fitness is better than Best Fitness then
32:       Update Best Solution and Best Fitness with this new solution
33:   end if
34:   Track Best Fitness across generations in fitness history
35: end for
36: Step 5: Return Results
37: Output: The best solution (selected features) with the highest fitness score,
    fitness history

```

Golden Ratio Method (GRM)

The Golden Ratio Method (GRM) is an optimization technique inspired by the golden ratio ($\phi \approx 1.618$), often observed in natural phenomena for efficient search and refinement. In GRM, candidate solutions represent points in a search space, and their positions are iteratively adjusted based on the golden ratio to converge toward optimal solutions.

Algorithm Mechanics GRM operates through the following phases:

- **Initialization:** A population of candidate solutions is generated randomly within defined bounds. Each candidate is represented by a binary vector where each bit indicates whether a feature is selected.
- **Golden Ratio-Based Search:** Each candidate solution's position is updated based on the golden ratio, using two reference points derived from the current solution and the best-known solution. This approach helps balance exploration and exploitation by guiding candidates towards promising regions.

- **Fitness Evaluation:** Each candidate is evaluated using a fitness function, such as cross-validated accuracy with a classifier, to determine the quality of the selected features.

The algorithm iteratively applies these steps until a termination criterion is met, such as reaching a maximum number of iterations.

Algorithm 8 Golden Ratio Method (GRM) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each candidate solution i in range (1, Population Size) **do**
- 3: Generate a random binary vector of length num_features, representing selected features
- 4: **end for**
- 5: **Step 2: Calculate Initial Fitness**
- 6: **for** each candidate solution i **do**
- 7: Evaluate fitness using cross-validation with XGBoost for each particle's selected features (see Equation 5)
- 8: Store fitness score for each candidate
- 9: **end for**
- 10: Identify the best initial solution as Best Solution with fitness Best Fitness
- 11: **Step 3: Optimization Loop Using Golden Ratio**
- 12: **for** each iteration in range (1, Max Iterations) **do**
- 13: **for** each candidate solution i **do**
- 14: Generate two new candidate points using the golden ratio ϕ :
 New Point 1 = Candidate + $\phi \times (\text{Best Solution} - \text{Candidate})$
 New Point 2 = Candidate - $\phi \times (\text{Best Solution} - \text{Candidate})$
- 15: Clip values in each new point to ensure they remain within bounds
- 16: Evaluate fitness for New Point 1 and New Point 2
- 13: **end for**
- 12: **end for**

```

17:         if either new point has a better fitness than the current solution then
18:             Update the current solution with the point of having the better
               fitness
19:         end if
20:     end for
21:     Update Best Solution and Best Fitness if a better solution is found in this
       iteration
22:     Track Best Fitness across iterations in fitness history
23: end for
24: Step 4: Return Results
25: Output: The best solution (selected features) with the highest fitness score,
       fitness history

```

Manta Ray Foraging Optimization (MRFO)

The Manta Ray Foraging Optimization (MRFO) is a metaheuristic algorithm inspired by manta rays' unique foraging behaviour, which includes coordinated movements for prey capture. In MRFO, candidate solutions represent individual manta rays, with each ray's position in the search space iteratively updated to explore and exploit potential solutions effectively.

Algorithm Mechanics MRFO operates through three primary foraging mechanisms:

- **Chain Foraging:** In this phase, manta rays move in a chain-like formation, where each ray updates its position based on the ray ahead or the current best solution. This behaviour promotes coordinated exploration across the

search space.

- **Cyclone Foraging:** Here, manta rays move in a spiralling pattern towards the best-known solution, allowing for a more localized search that refines promising areas of the solution space.
- **Somersault Foraging:** Manta rays perform a somersault around the best solution, enhancing exploration and promoting diversity by moving within a range around the best-known solution.

The algorithm iteratively applies these mechanisms until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a target fitness level.

Algorithm 9 Manta Ray Foraging Optimization (MRFO) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each manta ray i in range (1, Population Size) **do**
- 3: Generate a random binary vector of length num_features, representing selected features
- 4: **end for**
- 5: **Step 2: Calculate Initial Fitness**
- 6: **for** each manta ray i **do**
- 7: Evaluate fitness using cross-validation with XGBoost for each particle's selected features (see Equation 5)
- 8: Store fitness score for each manta ray
- 9: **end for**
- 10: Identify the best initial solution as Best Solution with fitness Best Fitness
- 11: **Step 3: Optimization Loop Using MRFO Mechanisms**
- 12: **for** each iteration in range (1, Max Iterations) **do**

```

13:   Chain Foraging: Each manta ray updates its position based on the
    preceding ray or the current best solution
14:   for each manta ray  $i$  do
15:       Update position based on chain movement towards the best solution or
    the position of ray  $i - 1$ 
16:   end for
17:   Cyclone Foraging: Each manta ray spirals towards the best-known
    solution
18:   for each manta ray  $i$  do
19:       Adjust position in a spiral pattern towards the best-known solution
20:   end for
21:   Somersault Foraging: Each manta ray explores within a radius around
    the best solution
22:   for each manta ray  $i$  do
23:       Update position based on somersault foraging around the best solution
24:   end for
25:   Evaluate fitness for all updated positions
26:   Update Best Solution and Best Fitness if a better solution is found in this
    iteration
27:   Track Best Fitness across iterations in fitness history
28: end for
29: Step 4: Return Results
30: Output: The best solution (selected features) with the highest fitness score,
    fitness history

```

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a popular metaheuristic inspired by the social behaviour of birds flocking or fish schooling. In PSO, candidate solutions are modelled as particles that "fly" through the search space, with each particle adjusting its position based on its personal experience and the collective experience of the swarm. This algorithm leverages both individual and social learning to converge toward optimal solutions.

Algorithm Mechanics PSO operates through three main steps:

- **Initialization:** A population of particles is randomly generated, each with an associated velocity and position representing a candidate solution.
- **Position and Velocity Update:** Each particle updates its velocity and position based on its own best-known position, the best-known position in its neighbourhood, and its current velocity. This balance allows for a mixture of exploration and exploitation in the search space.
- **Fitness Evaluation:** Each particle is evaluated based on a fitness function, often cross-validated accuracy or another metric, to determine the quality of the solution represented by its position.

The algorithm iterates these steps until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a desired fitness level.

Algorithm 10 Particle Swarm Optimization (PSO) for Feature Selection

1: **Step 1: Initialize Population**

```

2: for each particle  $i$  in range (1, Population Size) do
3:   Randomly generate a binary vector of length num_features representing
      selected features
4:   Initialize velocity for each particle as a vector of the same length
5:   Set each particle's personal best position to its initial position
6: end for
7: Identify the global best position Best Position with fitness Best Fitness from
      initial particles
8: Step 2: Optimization Loop
9: for each iteration in range (1, Max Iterations) do
10:   for each particle  $i$  do
11:     Update velocity using personal best, global best, and inertia terms:

        
$$\text{velocity}[i] = w \times \text{velocity}[i] + c_1 \times \text{rand}() \times (\text{personal best}[i] - \text{position}[i]) + c_2 \times \text{rand}() \times (\text{Best Position} - \text{position}[i])$$


12:     Update position based on the updated velocity:

        
$$\text{position}[i] = \text{position}[i] + \text{velocity}[i]$$


13:     Apply a sigmoid function to ensure binary feature selection, converting
        each dimension to 0 or 1
14:     Evaluate fitness using cross-validation with XGBoost for each particle's
        selected features (see Equation 5)
15:     if updated fitness is better than particle's personal best fitness then
16:       Update particle's personal best position and fitness
17:     end if
18:   end for
19:   Update global best position Best Position and fitness Best Fitness if a better
      solution is found in this iteration
20:   Track Best Fitness across iterations in fitness history

```

21: **end for**

22: **Step 3: Return Results**

23: **Output:** The best solution (selected features) with the highest fitness score,
fitness history

Walrus Optimization Algorithm (WaOA)

The Walrus Optimization Algorithm (WaOA) is a metaheuristic algorithm inspired by the social and foraging behaviors of walruses. In WaOA, candidate solutions represent individual walruses, which adapt their positions in the search space based on different behaviors, such as feeding, migration, and escaping/fighting. This structured behavior allows for effective exploration and exploitation to achieve optimal solutions.

Algorithm Mechanics WaOA operates through three main foraging mechanisms:

- **Feeding Phase:** Walruses move towards the best-known solution or promising areas to exploit high-quality solutions. This behavior promotes local search around successful solutions.
- **Migration Phase:** Walruses perform broader, random movements to explore the search space. This movement aims to prevent the algorithm from becoming trapped in local optima and to promote exploration of new areas.
- **Escaping/Fighting Phase:** Walruses make smaller, refining movements around high-quality solutions to balance between exploration and exploitation.

This helps refine promising solutions by focusing on adjustments within a local neighborhood.

The algorithm iteratively applies these mechanisms until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a target fitness level.

Algorithm 11 Walrus Optimization Algorithm (WaOA) for Feature Selection

```
1: Step 1: Initialize Population
2: for each walrus  $i$  in range (1, Population Size) do
3:   Generate a random binary vector of length num_features representing se-
     lected features
4: end for
5: Step 2: Calculate Initial Fitness
6: for each walrus  $i$  do
7:   Evaluate fitness using cross-validation with XGBoost for each walrus's
     selected features (see Equation 5)
8:   Store fitness score for each walrus
9: end for
10: Identify the best initial solution as Best Solution with fitness Best Fitness
11: Step 3: Optimization Loop Using WaOA Mechanisms
12: for each iteration in range (1, Max Iterations) do
13:   for each walrus  $i$  do
14:     Select a random number Phase Choice to determine the walrus's behav-
        ior
15:     if Phase Choice < 0.4 then
16:       Feeding Phase: Move walrus towards the best-known solution
17:       Update position using movement influenced by Best Solution
```

```

18:         else if  $0.4 \leq \text{Phase Choice} < 0.7$  then
19:             Migration Phase: Perform a broad random movement for explo-
                ration
20:             Update position by moving to a randomly selected walrus within the
                population
21:         else
22:             Escaping/Fighting Phase: Small refining movement around
                Best Solution
23:             Update position by making a small adjustment around Best Solution
24:         end if
25:         Ensure updated position remains within bounds and apply binary con-
            version to represent selected features
26:         Evaluate fitness for the updated position
27:         if updated fitness is better than current walrus's fitness then
28:             Update walrus's position and fitness
29:         end if
30:     end for
31:     Update Best Solution and Best Fitness if a better solution is found in this
        iteration
32:     Track Best Fitness across iterations in fitness history
33: end for
34: Step 4: Return Results
35: Output: The best solution (selected features) with the highest fitness score,
        fitness history

```

This pseudocode outlines the WaOA as a feature selection algorithm, focusing on structured phases that help balance exploration and exploitation, with evaluation via cross-validated XGBoost for accurate fitness assessment. The iterative application

of these distinct phases promotes robustness in feature selection by leveraging WaOA's unique foraging-inspired behaviors.

To conclude the discussion of standard metaheuristic algorithms, it's clear that each algorithm applied different optimization strategies to identify the most informative feature subsets. By exploring various approaches, these metaheuristics demonstrated the ability to effectively navigate large, complex search spaces, balancing exploration and exploitation to support improved wildfire prediction. However, despite their strengths, conventional metaheuristic methods face several limitations:

- **Escaping Local Optima:** Many standard algorithms, while robust, may struggle to consistently escape local optima in high-dimensional feature spaces, potentially limiting their effectiveness for complex, interdependent wildfire data Agrawal et al. [2021] .
- **Handling High Dimensionality:** As dimensionality increases, the computational load and convergence challenges also grow, impacting the scalability and overall performance of these methodsAkinola et al. [2022].

Given these limitations, a specialized approach for feature selection is advantageous. The Liver Cancer Algorithm (LCA) was selected for targeted improvement due to its adaptability to high-dimensional data and robust search capabilities. However, LCA has unique challenges, as highlighted in recent studies. Despite its strengths, the algorithm can exhibit limitations in precise refinement near optimal solutions, given its reliance on Levy Flight and Random Opposition-Based Learning (ROBL). These mechanisms, while enhancing exploration and diversity, may lead

to overshooting and can occasionally slow down convergence near optimal solutions due to their inherent randomness.

To address these issues, integrating a Spiral Update mechanism into LCA offers a systematic approach to enhance search precision. The Spiral Update guides the algorithm around promising solutions in a controlled path, providing more accurate, fine-tuned adjustments as it nears the optimal solution. This improvement enables LCA to more effectively balance exploration and exploitation, reducing the tendency to overshoot while maintaining a steady convergence path. In high-dimensional contexts such as wildfire prediction, the refined search offered by Spiral Update increases the precision and stability of the algorithm, positioning it to outperform conventional methods by reliably converging toward optimal solutions.

Liver Cancer Algorithm (LCA)

The Liver Cancer Algorithm (LCA) is a bio-inspired metaheuristic optimization algorithm that simulates the progression and mutation patterns observed in liver cancer cells to explore potential solutions in an optimization search space. In LCA, candidate solutions act as individual cells, with each cell representing a unique feature subset. The algorithm iteratively refines these cells, allowing them to "mutate" and explore different areas of the search space.

3.4.3 Mathematical Model of LCA

Tumor Growth

The mathematical model of the Liver Cancer Algorithm (LCA) is centered around simulating the tumor growth and metastasis process. The initial stage involves calculating the tumor's size, which is critical for setting up the algorithm's population. This process is modeled by:

$$\text{Position}_i^j = \frac{\pi}{6} (\text{length}_j \cdot \text{width}_j \cdot \text{height}_j) - (\text{lb} + (\text{ub} - \text{lb}) - r_d \cdot \text{Position}_i^j) \quad (3.7)$$

where Position_i^j represents the estimated volume of the tumor for the i -th agent in the j -th dimension, lb and ub are the lower and upper bounds of the search space, respectively, and r_d is a random number between 0 and 1 that facilitates exploration by introducing stochastic variation.

To model the growth of the tumor, the position size is updated dynamically using:

$$\text{Position} = \frac{\pi}{6} \cdot f \cdot (l \cdot w)^{3/2} \quad (3.8)$$

where $f = 1$ is a constant representing a specific tumor type, and l and w denote the length and width of the tumor, respectively. This formulation captures the geometric expansion behavior of the tumor, allowing the algorithm to balance exploration and exploitation effectively.

The initial size calculation ensures a diverse and well-distributed population,

enabling the algorithm to adaptively explore the search space during optimization.

Optimization Stages

Following the initial tumor size calculation, the next phase simulates the replication process observed in malignant tumors. This stage models the tumor's ability to duplicate itself in different regions within the liver, reflecting its invasive behavior.

The replication phase relies on the exponential growth model, which is commonly used to describe cellular division in cancers, including hepatocellular carcinoma. The tumor growth process is expressed mathematically as:

$$PG_i^j = \frac{dV}{dt} = r \times Position_t, \quad t \in [1 \dots T], \quad i \in [1 \dots N], \quad (3.9)$$

where PG represents the tumor growth rate, r denotes the radius of the tumor modeled as an ellipsoidal shape, and $Position_t$ is derived from the earlier initialization step. The parameters include: - T : Maximum number of iterations, - N : Population size or number of agents.

Advanced Search Dynamics

To improve exploration and avoid local optima, the replication phase integrates a Lévy flight-based search mechanism. The random walk process follows:

$$Lv(D) = 0.01 \times \frac{\text{rand}(1, D) \cdot \sigma}{|\text{rand}(1, D)|^{\frac{1}{\beta}}}, \quad (3.10)$$

where D is the dimension size, β controls the step distribution, and σ determines the scale factor. The scale σ is calculated as:

$$\sigma = \left(\frac{\Gamma(1 + \beta) \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}}, \quad (3.11)$$

where Γ denotes the Gamma function.

Exploration and Decision Mechanism. Based on the growth dynamics, each tumor evaluates its surroundings and determines its next position through the following steps:

$$y = Position + PG, \quad (3.12)$$

$$Z = Y + S \times LF(D), \quad (3.13)$$

where S represents random scaling factors to enhance exploration.

Finally, positions are updated based on fitness evaluations:

$$Position_{i,t+1} = \begin{cases} y, & \text{if } fit(y) < fit(Position_i), \\ z, & \text{if } fit(z) < fit(Position_i), \end{cases} \quad (3.14)$$

where $fit(\cdot)$ represents the fitness function used to evaluate solution quality.

In the final stage of the LCA, tumor progression mimics the metastatic behavior of cancer by applying genetic operators such as mutation and crossover.

Mutation: Mutation introduces diversity by randomly altering components of the tumor position. The mutation operator is defined as:

$$y_{Mut} = \begin{cases} Position & \text{if } rand_1 \geq \zeta \\ y & \text{else} \end{cases} \quad (3.15)$$

$$z_{Mut} = \begin{cases} Position & \text{if } rand_2 \geq \zeta \\ z & \text{else} \end{cases} \quad (3.16)$$

where:

$$\zeta = \frac{1}{T}, \quad y = |Position - Position_j|, \quad z = y - S \quad (3.17)$$

Crossover: Crossover combines elements from two parents to generate offspring. The new position after crossover is:

$$w_{\text{Cross}} = \tau \cdot y_{\text{Mut}} + (1 - \tau') \cdot z_{\text{Mut}}, \quad \tau \neq \tau' \quad (3.18)$$

Selection: A greedy selection process evaluates the fitness of each position, retaining the best solutions for the next iteration:

$$Position_{i,t+1} = \begin{cases} y_{\text{Mut}} & \text{if } fit(y_{\text{Mut}}) < fit(Position_i) \\ z_{\text{Mut}} & \text{if } fit(z_{\text{Mut}}) < fit(Position_i) \\ w_{\text{Cross}} & \text{if } fit(w_{\text{Cross}}) < fit(Position_i) \end{cases} \quad (3.19)$$

The mutation and crossover operations ensure sufficient exploration and exploitation within the search space, enabling the LCA algorithm to converge effectively towards optimal solutions.

Following the initialization, the LCA simulates tumor replication and spreading within the host organ, reflecting stages of disease progression. This adaptive search process is governed by:

$$Position = 6 \times f \times (\ell \cdot w)^{3/2} \quad (3.20)$$

where f is a constant reflecting specific tumor characteristics, and ℓ and w represent the evolving dimensions of the tumor within the search space.

The replication of the tumor, which signifies a critical phase of disease progression, is mathematically described by:

$$\frac{dV}{dt} = r \cdot \text{Position}_t, \quad t \in [1, T], \quad i \in [1, N] \quad (3.21)$$

where V is the volume of the tumor position, r defines the growth rate, T is the maximum number of iterations, and N represents the number of search agents.

To effectively simulate the metastatic spread, LCA utilizes Lévy flight dynamics, enhancing its exploratory capabilities across the problem landscape:

$$L(D) = 0.01 \times \text{rand}(1, D) \times \sigma \quad (3.22)$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta} \quad (3.23)$$

These equations govern the stochastic and strategic dispersal of search agents, mimicking the unpredictable and aggressive spread of liver cancer cells across the liver landscape, optimizing the algorithm's capability to locate and refine the best solutions for complex optimization problems.

3.4.4 Algorithm Mechanics

The Liver Cancer Algorithm (LCA) employs mutation-based processes inspired by tumor growth, replication, and metastasis to perform optimization. Its mechanics are as follows:

- **Primary Mutation:** This process introduces random changes in the position of candidate solutions, simulating the unpredictable growth of tumors. It promotes **diversity** within the population, ensuring broad exploration of the search space to avoid premature convergence.

- **Opposition-Based Mutation:** Inspired by opposition-based learning (ROBL), this mechanism generates complementary solutions by reflecting positions across their opposite values within the search bounds. This process expands the search area, allowing the algorithm to **escape local optima** and improve solution diversity.
- **Crossover Mutation:** Borrowing principles from genetic algorithms, LCA combines features of high-fitness solutions (parents) to produce offspring with **enhanced traits**. This crossover mechanism allows exploitation of promising areas while retaining diversity in the population.

The algorithm iteratively applies these mutation processes until a termination criterion, such as the maximum number of iterations, is met.

Algorithm 12 Liver Cancer Algorithm (LCA) for Feature Selection

```
1: Step 1: Initialization
2: Initialize population size  $N$  and maximum iterations  $T$ 
3: for each agent  $i$  in range  $(1, N)$  do
4:     Generate random positions  $\text{Position}_i$  using Eq. (3.7)
5: end for
6: Step 2: Fitness Evaluation
7: for each agent  $i$  do
8:     Evaluate fitness using cross-validation (e.g., XGBoost) for selected features
9:     Store fitness scores and identify Best Position and Best Fitness
10: end for
11: Step 3: Iterative Optimization Process
12: for each iteration  $t$  in range  $(1, T)$  do
13:     for each agent  $i$  do
14:         Compute fitness for current position
15:         if  $\text{Position} < 0.8$  or  $\text{Position} > 0.8$  then ▷ Exploitation Phase
16:             Update tumor position using Eq. (3.20)
17:         else if  $0.8 < \text{Position} < 1.96$  then ▷ Replication Phase
18:             Update position using Eq. (3.20) (Exponential Growth)
19:         else ▷ Exploration Phase
20:             Apply mutation using Eq. (3.15) and Eq. (3.16)
21:             Apply crossover using Eq. (3.18)
22:             Evaluate new positions using fitness and select the best
23:         end if
24:     end for
25:     Step 4: Update Population and Best Fitness
26:     Compare updated fitness values with Best Fitness
27:     if new fitness improves then
28:         Update Best Position and Best Fitness
29:     end if
30: end for
31: Step 5: Output Results
32: Return Best Position and Best Fitness
```

Manta Ray Foraging Optimization (MRFO)

The Manta Ray Foraging Optimization (MRFO) is a metaheuristic algorithm inspired by manta rays' unique foraging behaviour, which includes coordinated movements for prey capture. In MRFO, candidate solutions represent individual manta rays, with each ray's position in the search space iteratively updated to explore and exploit potential solutions effectively.

Algorithm Mechanics MRFO operates through three primary foraging mechanisms:

- **Chain Foraging:** In this phase, manta rays move in a chain-like formation, where each ray updates its position based on the ray ahead or the current best solution. This behaviour promotes coordinated exploration across the search space.
- **Cyclone Foraging:** Here, manta rays move in a spiralling pattern towards the best-known solution, allowing for a more localized search that refines promising areas of the solution space.
- **Somersault Foraging:** Manta rays perform a somersault around the best solution, enhancing exploration and promoting diversity by moving within a range around the best-known solution.

The algorithm iteratively applies these mechanisms until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a target fitness level.

Algorithm 13 Manta Ray Foraging Optimization (MRFO) for Feature Selection

```
1: Step 1: Initialize Population
2: for each manta ray  $i$  in range (1, Population Size) do
3:   Generate a random binary vector of length num_features, representing
     selected features
4: end for
5: Step 2: Calculate Initial Fitness
6: for each manta ray  $i$  do
7:   Evaluate fitness using cross-validation with XGBoost for each particle's
     selected features (see Equation 5)
8:   Store fitness score for each manta ray
9: end for
10: Identify the best initial solution as Best Solution with fitness Best Fitness
11: Step 3: Optimization Loop Using MRFO Mechanisms
12: for each iteration in range (1, Max Iterations) do
13:   Chain Foraging: Each manta ray updates its position based on the
     preceding ray or the current best solution
14:   for each manta ray  $i$  do
15:     Update position based on chain movement towards the best solution or
     the position of ray  $i - 1$ 
16:   end for
17:   Cyclone Foraging: Each manta ray spirals towards the best-known
     solution
18:   for each manta ray  $i$  do
19:     Adjust position in a spiral pattern towards the best-known solution
20:   end for
```

```

21:   Somersault Foraging: Each manta ray explores within a radius around
    the best solution
22:   for each manta ray  $i$  do
23:       Update position based on somersault foraging around the best solution
24:   end for
25:   Evaluate fitness for all updated positions
26:   Update Best Solution and Best Fitness if a better solution is found in this
    iteration
27:   Track Best Fitness across iterations in fitness history
28: end for
29: Step 4: Return Results
30: Output: The best solution (selected features) with the highest fitness score,
    fitness history

```

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a popular metaheuristic inspired by the social behaviour of birds flocking or fish schooling. In PSO, candidate solutions are modelled as particles that "fly" through the search space, with each particle adjusting its position based on its personal experience and the collective experience of the swarm. This algorithm leverages both individual and social learning to converge toward optimal solutions.

Algorithm Mechanics PSO operates through three main steps:

- **Initialization:** A population of particles is randomly generated, each with an associated velocity and position representing a candidate solution.

- **Position and Velocity Update:** Each particle updates its velocity and position based on its own best-known position, the best-known position in its neighbourhood, and its current velocity. This balance allows for a mixture of exploration and exploitation in the search space.
- **Fitness Evaluation:** Each particle is evaluated based on a fitness function, often cross-validated accuracy or another metric, to determine the quality of the solution represented by its position.

The algorithm iterates these steps until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a desired fitness level.

Algorithm 14 Particle Swarm Optimization (PSO) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each particle i in range $(1, \text{Population Size})$ **do**
- 3: Randomly generate a binary vector of length `num_features` representing selected features
- 4: Initialize velocity for each particle as a vector of the same length
- 5: Set each particle's personal best position to its initial position
- 6: **end for**
- 7: Identify the global best position `Best Position` with fitness `Best Fitness` from initial particles
- 8: **Step 2: Optimization Loop**
- 9: **for** each iteration in range $(1, \text{Max Iterations})$ **do**
- 10: **for** each particle i **do**
- 11: Update velocity using personal best, global best, and inertia terms:

$$\text{velocity}[i] = w \times \text{velocity}[i] + c_1 \times \text{rand}() \times (\text{personal best}[i] - \text{position}[i]) + c_2 \times \text{rand}() \times (\text{Best Position} - \text{position}[i])$$

```

12:         Update position based on the updated velocity:

           position[i] = position[i] + velocity[i]

13:         Apply a sigmoid function to ensure binary feature selection, converting
           each dimension to 0 or 1

14:         Evaluate fitness using cross-validation with XGBoost for each particle's
           selected features (see Equation 5)

15:         if updated fitness is better than particle's personal best fitness then
16:             Update particle's personal best position and fitness
17:         end if
18:     end for

19:     Update global best position Best Position and fitness Best Fitness if a better
       solution is found in this iteration

20:     Track Best Fitness across iterations in fitness history

21: end for

22: Step 3: Return Results

23: Output: The best solution (selected features) with the highest fitness score,
       fitness history

```

Walrus Optimization Algorithm (WaOA)

The Walrus Optimization Algorithm (WaOA) is a metaheuristic algorithm inspired by the social and foraging behaviors of walruses. In WaOA, candidate solutions represent individual walruses, which adapt their positions in the search space based on different behaviors, such as feeding, migration, and escaping/fighting. This structured behavior allows for effective exploration and exploitation to achieve optimal solutions.

Algorithm Mechanics WaOA operates through three main foraging mechanisms:

- **Feeding Phase:** Walruses move towards the best-known solution or promising areas to exploit high-quality solutions. This behavior promotes local search around successful solutions.
- **Migration Phase:** Walruses perform broader, random movements to explore the search space. This movement aims to prevent the algorithm from becoming trapped in local optima and to promote exploration of new areas.
- **Escaping/Fighting Phase:** Walruses make smaller, refining movements around high-quality solutions to balance between exploration and exploitation. This helps refine promising solutions by focusing on adjustments within a local neighborhood.

The algorithm iteratively applies these mechanisms until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a target fitness level.

Algorithm 15 Walrus Optimization Algorithm (WaOA) for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each walrus i in range $(1, \text{Population Size})$ **do**
- 3: Generate a random binary vector of length `num_features` representing selected features
- 4: **end for**
- 5: **Step 2: Calculate Initial Fitness**
- 6: **for** each walrus i **do**

```

7:    Evaluate fitness using cross-validation with XGBoost for each walrus's
    selected features (see Equation 5)
8:    Store fitness score for each walrus
9: end for
10: Identify the best initial solution as Best Solution with fitness Best Fitness
11: Step 3: Optimization Loop Using WaOA Mechanisms
12: for each iteration in range (1, Max Iterations) do
13:     for each walrus  $i$  do
14:         Select a random number Phase Choice to determine the walrus's behav-
        ior
15:         if Phase Choice < 0.4 then
16:             Feeding Phase: Move walrus towards the best-known solution
17:             Update position using movement influenced by Best Solution
18:         else if  $0.4 \leq$  Phase Choice < 0.7 then
19:             Migration Phase: Perform a broad random movement for explo-
            ration
20:             Update position by moving to a randomly selected walrus within the
            population
21:         else
22:             Escaping/Fighting Phase: Small refining movement around
            Best Solution
23:             Update position by making a small adjustment around Best Solution
24:         end if
25:         Ensure updated position remains within bounds and apply binary con-
        version to represent selected features
26:         Evaluate fitness for the updated position
27:         if updated fitness is better than current walrus's fitness then

```

```

28:         Update walrus's position and fitness
29:     end if
30: end for
31: Update Best Solution and Best Fitness if a better solution is found in this
    iteration
32: Track Best Fitness across iterations in fitness history
33: end for
34: Step 4: Return Results
35: Output: The best solution (selected features) with the highest fitness score,
    fitness history

```

This pseudocode outlines the WaOA as a feature selection algorithm, focusing on structured phases that help balance exploration and exploitation, with evaluation via cross-validated XGBoost for accurate fitness assessment. The iterative application of these distinct phases promotes robustness in feature selection by leveraging WaOA's unique foraging-inspired behaviors.

To conclude the discussion of standard metaheuristic algorithms, it's clear that each algorithm applied different optimization strategies to identify the most informative feature subsets. By exploring various approaches, these metaheuristics demonstrated the ability to effectively navigate large, complex search spaces, balancing exploration and exploitation to support improved wildfire prediction. However, despite their strengths, conventional metaheuristic methods face several limitations:

- **Escaping Local Optima:** Many standard algorithms, while robust, may struggle to consistently escape local optima in high-dimensional feature spaces,

potentially limiting their effectiveness for complex, interdependent wildfire data Agrawal et al. [2021] .

- **Handling High Dimensionality:** As dimensionality increases, the computational load and convergence challenges also grow, impacting the scalability and overall performance of these methodsAkinola et al. [2022].

Given these limitations, a specialized approach for feature selection is advantageous. The Liver Cancer Algorithm (LCA) was selected for targeted improvement due to its adaptability to high-dimensional data and robust search capabilities. However, LCA has unique challenges, as highlighted in recent studies. Despite its strengths, the algorithm can exhibit limitations in precise refinement near optimal solutions, given its reliance on Levy Flight and Random Opposition-Based Learning (ROBL). These mechanisms, while enhancing exploration and diversity, may lead to overshooting and can occasionally slow down convergence near optimal solutions due to their inherent randomness.

To address these issues, integrating a Spiral Update mechanism into LCA offers a systematic approach to enhance search precision. The Spiral Update guides the algorithm around promising solutions in a controlled path, providing more accurate, fine-tuned adjustments as it nears the optimal solution. This improvement enables LCA to more effectively balance exploration and exploitation, reducing the tendency to overshoot while maintaining a steady convergence path. In high-dimensional contexts such as wildfire prediction, the refined search offered by Spiral Update increases the precision and stability of the algorithm, positioning it to outperform conventional methods by reliably converging toward optimal solutions.

Given these limitations, a specialized approach for feature selection is advantageous. The Liver Cancer Algorithm (LCA) was selected for targeted improvement

due to its adaptability to high-dimensional data and robust search capabilities. However, LCA has unique challenges, as highlighted in recent studies. Despite its strengths, the algorithm can exhibit limitations in precise refinement near optimal solutions, given its reliance on Levy Flight and Random Opposition-Based Learning (ROBL). These mechanisms, while enhancing exploration and diversity, may lead to overshooting and can occasionally slow down convergence near optimal solutions due to their inherent randomness.

To address these issues, integrating a Spiral Update mechanism into LCA offers a systematic approach to enhance search precision. The Spiral Update guides the algorithm around promising solutions in a controlled path, providing more accurate, fine-tuned adjustments as it nears the optimal solution. This improvement enables LCA to more effectively balance exploration and exploitation, reducing the tendency to overshoot while maintaining a steady convergence path. In high-dimensional contexts such as wildfire prediction, the refined search offered by Spiral Update increases the precision and stability of the algorithm, positioning it to outperform conventional methods by reliably converging toward optimal solutions.

3.5 Proposed Algorithm

3.5.1 Spiral Implementation

Spiral structures generally appear in forms such as galaxies, hurricanes, seashells, and plant growth patterns. A consistent yet dynamic progression marks these natural phenomena, inspiring optimization techniques that mimic their structured navigation through complex environments. The convergent nature of spirals,

coupled with their orderly expansion, makes them highly effective in optimization. They offer a path that balances thorough search space exploration with a steady movement towards optimal solutions (Omar et al. [2022], Tamura and Yasuda [2011]).

The spiral’s rotating and inward-curving trajectory enables smooth transitions toward high-potential areas without the abrupt jumps often seen in random exploration methods Polezhaev [2019]. This structured progression supports controlled convergence, particularly in high-dimensional and complex search spaces. Specifically, spiral dynamics provides the following advantages in optimization:

- **Improved Search Precision:** The gradual approach of the spiral path allows algorithms to fine-tune adjustments as they get closer to the optimal solution. This precise control helps reduce the risk of overshooting high-potential regions, addressing a common limitation in methods such as Levy flights.
- **Enhanced Exploration and Exploitation Balance:** Through systematic rotation throughout the search space, spiral dynamics strike a balance between exploring new regions and exploiting promising ones. This balanced approach is essential in optimization, preventing premature convergence and ensuring a comprehensive search space exploration.
- **Adaptability to Boundaries:** Spiral paths can be adjusted to remain within defined boundaries, making them particularly suitable for constrained optimization tasks. This flexibility allows the spiral dynamics to operate effectively in diverse problems where boundary conditions must be respected.

In optimization, the spiral dynamic’s predictable yet flexible path offers substan-

tial benefits for navigating high-dimensional spaces with complex interdependencies. For algorithms such as the Liver Cancer Algorithm (LCA), integrating a Spiral Update mechanism introduces a systematic search pattern that enhances precision in convergence. This enhancement is especially valuable in complex domains such as wildfire prediction, where efficient exploration and accurate targeting of high-impact feature subsets are critical to improve predictive performance.

Parameter Definitions

To formalize the implementation of various spiral types, the following parameters are consistently used throughout their mathematical representations:

- a : A scaling constant determining the spiral's initial size or growth rate.
- b : A constant controlling the spacing or rate of growth between successive loops of the spiral.
- φ : The polar angle, measured in radians, represents the rotational component of the spiral.
- θ : The angular increment, a step size determining the resolution of the spiral path.
- r : The radius or distance from the origin to a point on the spiral.
- v : A constant velocity used in some formulations to integrate linear motion with rotational dynamics.
- ω : The angular velocity defines the rate at which the spiral rotates about the origin.

These parameters are adapted for each specific spiral type to achieve tailored exploration and exploitation dynamics.

We utilized several spiral path types to implement the Spiral Update in the Liver Cancer Algorithm (LCA), each tailored to different optimization strategies. By incorporating these specific spiral dynamics into the LCA, we developed a more refined approach to feature selection, particularly well-suited for high-dimensional and interdependent datasets. Below, we outline each spiral type, its mathematical formulation, and its unique contributions to enhancing the LCA's performance.

The Archimedean Spiral

- **Archimedean Spiral:** The Archimedean spiral, named after the Greek mathematician Archimedes, describes a path in which a point moves outward from a fixed origin at a constant speed while rotating with a constant angular velocity. This type of spiral can be mathematically represented in polar coordinates by the formula:

$$r = b \cdot \theta \tag{3.24}$$

where r is the radius (distance from the origin), θ is the angle, and b is a constant controlling the spacing between loops. A higher value of b increases the distance between successive spiral turns.

The Archimedean spiral is beneficial in optimization, especially for systematically exploring search spaces. Because the radial distance r grows proportionally with θ , the spiral expands evenly from the origin. This steady spacing enables controlled and incremental searches, avoiding large jumps and ensuring consistent coverage across the search space.

Derivation of the Archimedean Spiral Formula To derive this formula, consider a point moving at constant velocity v along the x -axis in Cartesian coordinates. If the xy -plane rotates around the z -axis with a constant angular velocity ω , then the position of the point at any time t is given by:

$$x = (vt + c) \cos(\omega t) \quad \text{and} \quad y = (vt + c) \sin(\omega t) \quad (3.25)$$

where c is an initial radial offset. This formulation integrates linear motion with rotational movement, creating a spiral path over time. In polar coordinates, this can be expressed as:

$$r = \frac{v}{\omega} \cdot \theta + c \quad (3.26)$$

Key Characteristics

- **Constant Separation Distance:** Each loop of the Archimedean spiral is separated by a constant distance along any ray from the origin. For θ measured in radians, this spacing is $2\pi b$.
- **Uniform Expansion:** For large θ , movement along the spiral approximates uniform acceleration, providing a stable path ideal for controlled, predictable searches.
- **Symmetry:** The Archimedean spiral is symmetrical, with two mirrored arms—one for $\theta > 0$ and one for $\theta < 0$ —supporting balanced search patterns.

Algorithm 16 Liver Cancer Algorithm (LCA) with Archimedean Spiral Update.

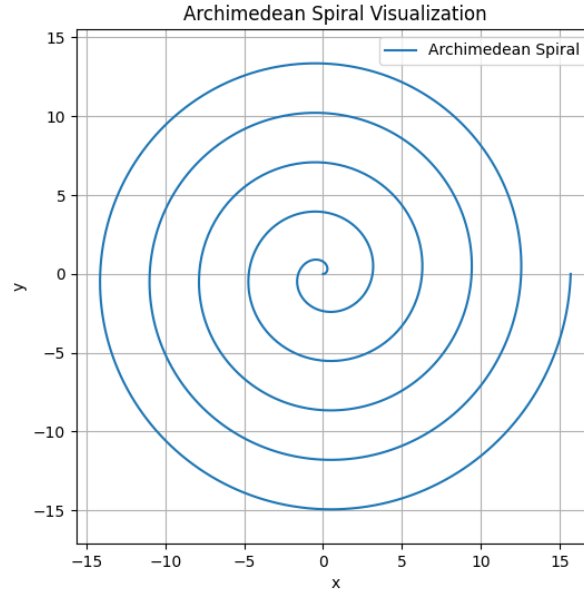


Figure 3.2: Visualization of the Archimedean Spiral

- 1: **Step 1: Initialize Population**
- 2: **for** each agent i in range $(0, \text{SearchAgents})$ **do**
- 3: Generate a binary vector of length dimension for feature selection
- 4: Generate opposite agents to enhance diversity and combine with initial agents
- 5: **end for**
- 6: **Step 2: Calculate Initial Fitness**
- 7: **for** each agent i **do**
- 8: Select features based on agent i 's binary vector
- 9: Evaluate fitness using cross-validation with XGBoost on the selected features
- 10: Store fitness score in $\text{fit}[i]$
- 11: **end for**
- 12: Identify the best initial solution, updating best_score and best_features

13: **Step 3: Optimization Loop**

14: **for** each iteration t in range (1, Max_iterations) **do**

15: **for** each agent i **do**

16: **Exploration with Levy Flight:** With probability 0.5, apply Levy flight to diversify the agent's position

17: **if** random() \leq 0.5 **then**

18: Generate a step using Levy flight and adjust the agent's position

19: **end if**

20: ****Archimedean Spiral Update**:** With probability 0.5, apply

21: **if** random() \leq 0.5 **then**

22: Set best_agent $\leftarrow X[\text{argmax}(\text{fit})]$

23: Compute the direction vector direction = best_agent - $X[i]$

24: Calculate the Euclidean norm of the direction vector:

$$\text{norm} = \|\text{direction}\|$$

25: **if** norm = 0 **then**

26: Set norm = 1 to avoid division by zero

27: **end if**

28: Normalize the direction vector:

$$\text{unit_direction} = \frac{\text{direction}}{\text{norm}}$$

29: Set the angular increment θ (e.g., $\theta = 0.1$)

30: Calculate the spiral step size based on constant b :

$$\text{step} = b \cdot \theta \cdot \text{unit_direction}$$

31: Update agent's position in the direction of the spiral step:

$$\text{new_position} = X[i] + \text{step}$$

```

32:         Convert updated position to binary by setting values above 0.5 to 1
           and others to 0:

$$X[i] = (\text{new\_position} > 0.5)$$


33:     end if
34:     Crossover and Mutation:
35:     Select a partner agent at random
36:     Apply crossover to generate a child:

$$\text{child} \leftarrow \text{crossover}(X[i], X[\text{partner\_idx}])$$


37:     Apply mutation to the child:

$$\text{mutated\_child} \leftarrow \text{mutation}(\text{child}, \text{mutation\_rate})$$


38:     Calculate the fitness of the mutated child
39:     if mutated_child fitness  $\geq$  current agent fitness then
40:         Replace the current agent with the mutated child
41:     end if
42: end for
43: Update Best Solution:
44: if Current best fitness is greater than previous best then
45:     Update best_score and best_features
46: end if
47: end for
48: Step 4: Return Results
49: Output: The best solution best_features, best fitness score best_score, and
           fitness history over iterations

```

The Euler Spiral

- **Euler Spiral:** The Euler spiral, also known as the Clothoid or Cornu spiral, is defined by a variable curvature that increases linearly with the arc length. As the spiral progresses, its curvature continuously changes, allowing for smooth transitions in direction Levien [2008], Seggern [1994], Cherchi et al. [2013]. In polar coordinates, the Euler spiral does not have a straightforward formula like the Archimedean spiral but can be described parametrically using Fresnel integrals:

$$x(t) = C(t) = \int_0^t \cos\left(\frac{\pi u^2}{2}\right) du \quad (3.27)$$

$$y(t) = S(t) = \int_0^t \sin\left(\frac{\pi u^2}{2}\right) du \quad (3.28)$$

where $C(t)$ and $S(t)$ are the Fresnel integrals, which describe the coordinates of the point on the spiral.

The Euler spiral is handy for optimizing smooth exploration. Its linearly changing curvature provides a gradual transition in direction, preventing abrupt movements. This helps navigate complex search spaces while maintaining a controlled exploration path.

Key Characteristics

- **Smooth Curvature Transition:** The Euler spiral's curvature varies linearly, allowing the search process to transition through potential solutions without abrupt changes smoothly.

- **Controlled Expansion:** As the curvature gradually increases, the spiral expands smoothly, ensuring the algorithm does not jump too rapidly between points.
- **Balanced Exploration and Exploitation:** The continuous curvature adjustment allows for a natural balance between exploration (diverging outward) and exploitation (converging inward), making it suitable for high-dimensional spaces.

Algorithm 17 Liver Cancer Algorithm (LCA) with Detailed Euler Spiral Update for Feature Selection

```

1: Step 1: Initialize Population
2: for each agent  $i$  in range  $(0, \text{SearchAgents})$  do
3:   Generate a binary vector of length dimension for feature selection
4:   Generate opposite agents to enhance diversity and combine with initial
   agents
5: end for
6: Step 2: Calculate Initial Fitness
7: for each agent  $i$  do
8:   Select features based on agent  $i$ 's binary vector
9:   Evaluate fitness using cross-validation with XGBoost on the selected features
10:  Store fitness score in  $\text{fit}[i]$ 
11: end for
12: Identify the best initial solution, updating  $\text{best\_score}$  and  $\text{best\_features}$ 
13: Step 3: Optimization Loop
14: for each iteration  $t$  in range  $(1, \text{Max\_iterations})$  do
15:   for each agent  $i$  do
16:     Exploration with Levy Flight: With probability 0.5, apply Levy
     flight to diversify the agent's position

```

```

17:      if random() < 0.5 then
18:          Generate a step using Levy flight and adjust the agent's position
19:      end if
20:      Euler Spiral Update: With probability 0.5, apply Euler spiral update
        towards the best agent
21:      if random() < 0.5 then
22:          Set best_agent  $\leftarrow X[\text{argmax}(\text{fit})]$ 
23:          Detailed Logic of Euler Spiral Update:
24:          Compute the direction vector  $\text{direction} = \text{best\_agent} - X[i]$ 
25:          Calculate the Euclidean norm of the direction vector:

                                
$$\text{norm} = \|\text{direction}\|$$


26:          if norm = 0 then
27:              Set norm = 1 to avoid division by zero
28:          end if
29:          Normalize the direction vector:

                                
$$\text{unit\_direction} = \frac{\text{direction}}{\text{norm}}$$


30:          Set the initial curvature and curvature increment:

                                
$$\text{curvature} = \text{curvature\_start} + \text{curvature\_increment} \cdot \text{norm}$$


31:          Calculate the Euler spiral step based on the curvature:

                                
$$\text{step} = \text{step\_size} \cdot \text{curvature} \cdot \text{unit\_direction}$$


32:          Update agent's position in the direction of the Euler spiral step:

                                
$$\text{new\_position} = X[i] + \text{step}$$


```

```

33:         Apply dynamic thresholding to convert the position to binary:

         $X[i] = (\text{new\_position} > \text{dynamic\_threshold}(t, \text{Max\_iterations})).\text{astype}(\text{int})$ 

34:     end if
35:     Crossover and Mutation:
36:     Select a partner agent at random
37:     Apply crossover to generate a child:

         $\text{child} \leftarrow \text{crossover}(X[i], X[\text{partner\_idx}])$ 

38:     Apply mutation to the child:

         $\text{mutated\_child} \leftarrow \text{mutation}(\text{child}, \text{mutation\_rate})$ 

39:     Calculate the fitness of the mutated child
40:     if mutated_child fitness  $\geq$  current agent fitness then
41:         Replace the current agent with the mutated child
42:     end if
43: end for
44: Update Best Solution:
45:     if Current best fitness is greater than previous best then
46:         Update best_score and best_features
47:     end if
48: end for
49: Step 4: Return Results
50: Output: The best solution best_features, best fitness score best_score, and
    fitness history over iterations

```

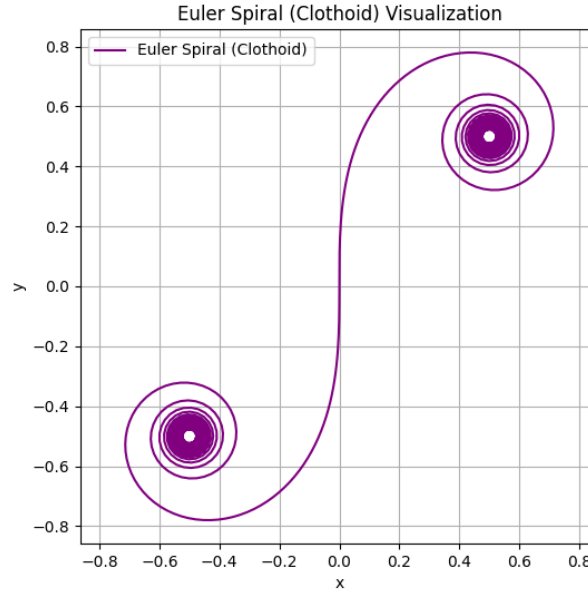


Figure 3.3: Visualization of the Euler Spiral

The Fermat Spiral

- **Fermat Spiral:** The Fermat spiral, or parabolic spiral, is characterized by the property that the area between consecutive turns remains constant while the distance between turns grows proportionally with the square root of the angle from the center. Unlike the Archimedean spiral, which maintains constant spacing, the Fermat spiral expands outward slower, making it suitable for controlled exploration with a gradual increase in search radius.

In polar coordinates, the Fermat spiral can be represented as:

$$r = \pm a\sqrt{\varphi} \quad (3.29)$$

where r is the radius (distance from the origin), φ is the angle, and a is a constant that controls the growth rate. The two signs produce two symmetrical arms of the spiral, both converging at the origin.

The Fermat spiral’s unique property of spacing growth makes it particularly effective in feature selection tasks for wildfire prediction. It allows for a structured, outward exploration. By gradually expanding the search radius, the Fermat spiral provides comprehensive coverage of the feature space, reducing the likelihood of missing important feature subsets.

Key Characteristics

- **Constant Area between Turns:** The Fermat spiral maintains a constant area between each pair of successive loops, supporting even exploration throughout the search space.
- **Gradual Expansion:** The distance between loops increases proportionally with $\sqrt{\varphi}$, allowing for controlled expansion that is particularly beneficial in high-dimensional search spaces.
- **Symmetry:** The two symmetrical branches of the Fermat spiral provide balanced coverage, supporting consistent exploration across the search space.

Algorithm 18 Liver Cancer Algorithm (LCA) with Detailed Fermat Spiral Update for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each agent i in range $(0, \text{SearchAgents})$ **do**
- 3: Generate a binary vector of length dimension for feature selection
- 4: Generate opposite agents to enhance diversity and combine with initial agents
- 5: **end for**
- 6: **Step 2: Calculate Initial Fitness**
- 7: **for** each agent i **do**

```

8:   Select features based on agent  $i$ 's binary vector
9:   Evaluate fitness using cross-validation with XGBoost on the selected features
10:  Store fitness score in  $\text{fit}[i]$ 
11: end for
12: Identify the best initial solution, updating  $\text{best\_score}$  and  $\text{best\_features}$ 
13: Step 3: Optimization Loop
14: for each iteration  $t$  in range  $(1, \text{Max\_iterations})$  do
15:   for each agent  $i$  do
16:     Exploration with Levy Flight: With probability 0.5, apply Levy
      flight to diversify the agent's position
17:     if  $\text{random}() \leq 0.5$  then
18:       Generate a step using Levy flight and adjust the agent's position
19:     end if
20:     Fermat Spiral Update: With probability 0.5, apply Fermat spiral
      update towards the best agent
21:     if  $\text{random}() \leq 0.5$  then
22:       Set  $\text{best\_agent} \leftarrow X[\text{argmax}(\text{fit})]$ 
23:       Detailed Logic of Fermat Spiral Update:
24:       Compute the direction vector  $\text{direction} = \text{best\_agent} - X[i]$ 
25:       Calculate the Euclidean norm of the direction vector:
          
$$\text{norm} = \|\text{direction}\|$$

26:       if  $\text{norm} = 0$  then
27:         Set  $\text{norm} = 1$  to avoid division by zero
28:       end if
29:       Normalize the direction vector:

```

$$\text{unit_direction} = \frac{\text{direction}}{\text{norm}}$$

30: Set the angle increment φ and calculate the Fermat spiral step size:

$$\text{step} = a \cdot \sqrt{\varphi} \cdot \text{unit_direction}$$

31: Update agent's position in the direction of the Fermat spiral step:

$$\text{new_position} = X[i] + \text{step}$$

32: Convert updated position to binary based on a dynamic threshold:

$$X[i] = (\text{new_position} > \text{dynamic_threshold}(t, \text{Max_iterations})).\text{astype}(\text{int})$$

33: **end if**

34: **Crossover and Mutation:**

35: Select a partner agent at random

36: Apply crossover to generate a child:

$$\text{child} \leftarrow \text{crossover}(X[i], X[\text{partner_idx}])$$

37: Apply mutation to the child:

$$\text{mutated_child} \leftarrow \text{mutation}(\text{child}, \text{mutation_rate})$$

38: Calculate the fitness of the mutated child

39: **if** mutated_child fitness \geq current agent fitness **then**

40: Replace the current agent with the mutated child

41: **end if**

42: **end for**

43: **Update Best Solution:**

44: **if** Current best fitness is greater than previous best **then**

45: Update best_score and best_features

46: **end if**

47: **end for**

48: **Step 4: Return Results**

49: **Output:** The best solution `best_features`, best fitness score `best_score`, and fitness history over iterations

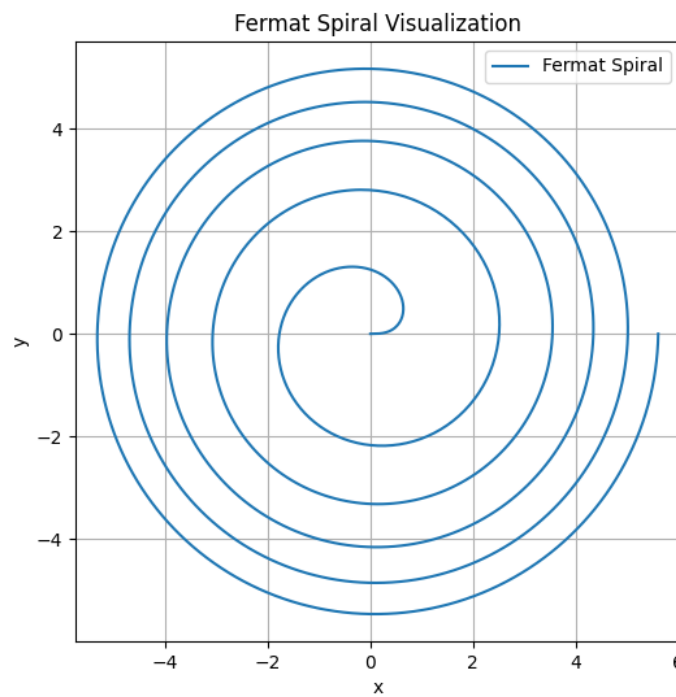


Figure 3.4: Visualization of the Fermat Spiral

The Golden Spiral

- **Golden Spiral:** The Golden spiral is a specific logarithmic spiral where the radius grows by a factor of the golden ratio $\varphi \approx 1.618$ for every quarter turn. This unique property of expansion enables a consistent scaling as the spiral moves outward, with each quarter turn moving further away from the

origin by a constant proportion. The Golden spiral is especially suitable for optimization tasks requiring smooth, proportional search space exploration.

In polar coordinates, the Golden spiral is represented by:

$$r = ae^{b\theta} \quad (3.30)$$

where r is the radius, θ is the angle, a is the initial radius, and b is the growth rate determined by the golden ratio, given by $b = \frac{\ln \varphi}{\pi/2}$.

In the context of feature selection for wildfire prediction, the Golden Spiral's proportional and smooth expansion systematically explores the feature space. This exploration ensures that the search process does not overlook critical regions, enhancing the likelihood of identifying optimal feature subsets that improve predictive performance.

Key Characteristics

- **Proportional Growth:** Each quarter turn increases the radius by a factor of φ , supporting systematic exploration across varying scales in the search space.
- **Smooth Expansion:** The Golden spiral grows outward smoothly, which is advantageous for controlled expansion in high-dimensional feature selection tasks.
- **Self-Similarity:** The self-similar nature of the Golden spiral supports consistent coverage across the search space, balancing both global exploration and local exploitation.

Algorithm 19 Liver Cancer Algorithm (LCA) with Detailed Golden Spiral Update for Feature Selection

```

1: Step 1: Initialize Population
2: for each agent  $i$  in range (0, SearchAgents) do
3:     Generate a binary vector of length dimension for feature selection
4:     Generate opposite agents to enhance diversity and combine with initial
        agents
5: end for
6: Step 2: Calculate Initial Fitness
7: for each agent  $i$  do
8:     Select features based on agent  $i$ 's binary vector
9:     Evaluate fitness using cross-validation with XGBoost on the selected features
10:    Store fitness score in fit[ $i$ ]
11: end for
12: Identify the best initial solution, updating best_score and best_features
13: Step 3: Optimization Loop
14: for each iteration  $t$  in range (1, Max_iterations) do
15:     for each agent  $i$  do
16:         Exploration with Levy Flight: With probability 0.5, apply Levy
            flight to diversify the agent's position
17:         if random()  $\leq$  0.5 then
18:             Generate a step using Levy flight and adjust the agent's position
19:         end if
20:         Golden Spiral Update: With probability 0.5, apply Golden spiral
            update towards the best agent
21:         if random()  $\leq$  0.5 then
22:             Set best_agent  $\leftarrow X[\text{argmax}(\text{fit})]$ 
23:             Detailed Logic of Golden Spiral Update:
24:             Compute the direction vector direction = best_agent -  $X[i]$ 

```

25: Calculate the Euclidean norm of the direction vector:

$$\text{norm} = \|\text{direction}\|$$

26: **if** norm = 0 **then**

27: Set norm = 1 to avoid division by zero

28: **end if**

29: Normalize the direction vector:

$$\text{unit_direction} = \frac{\text{direction}}{\text{norm}}$$

30: Set the angular increment θ and calculate the Golden spiral step size
using:

$$\text{step} = ae^{b\theta} \cdot \text{unit_direction}$$

31: Update agent's position in the direction of the Golden spiral step:

$$\text{new_position} = X[i] + \text{step}$$

32: Convert updated position to binary based on a dynamic threshold:

$$X[i] = (\text{new_position} > \text{dynamic_threshold}(t, \text{Max_iterations})).\text{astype}(\text{int})$$

33: **end if**

34: **Crossover and Mutation:**

35: Select a partner agent at random

36: Apply crossover to generate a child:

$$\text{child} \leftarrow \text{crossover}(X[i], X[\text{partner_idx}])$$

37: Apply mutation to the child:

$$\text{mutated_child} \leftarrow \text{mutation}(\text{child}, \text{mutation_rate})$$


```

38:         Calculate the fitness of the mutated child
39:         if mutated_child fitness > current agent fitness then
40:             Replace the current agent with the mutated child
41:         end if
42:     end for
43:     Update Best Solution:
44:     if Current best fitness is greater than previous best then
45:         Update best_score and best_features
46:     end if
47: end for
48: Step 4: Return Results
49: Output: The best solution best_features, best fitness score best_score, and
           fitness history over iterations

```

The Hyperbolic Spiral

- **Hyperbolic Spiral:** The hyperbolic spiral, also called the reciprocal spiral, is defined by a unique property where its radius decreases inversely with the angle. This creates a curve that approaches an asymptotic line as it extends, with a pitch angle that increases with distance from the origin. Unlike logarithmic or Archimedean spirals, the hyperbolic spiral's shape supports a decelerated outward exploration, which is particularly advantageous in controlled optimization tasks Bowser [1910], Øyvind Hammer [2016].

In polar coordinates, the hyperbolic spiral is represented by:

$$r = \frac{a}{\varphi} \quad (3.31)$$

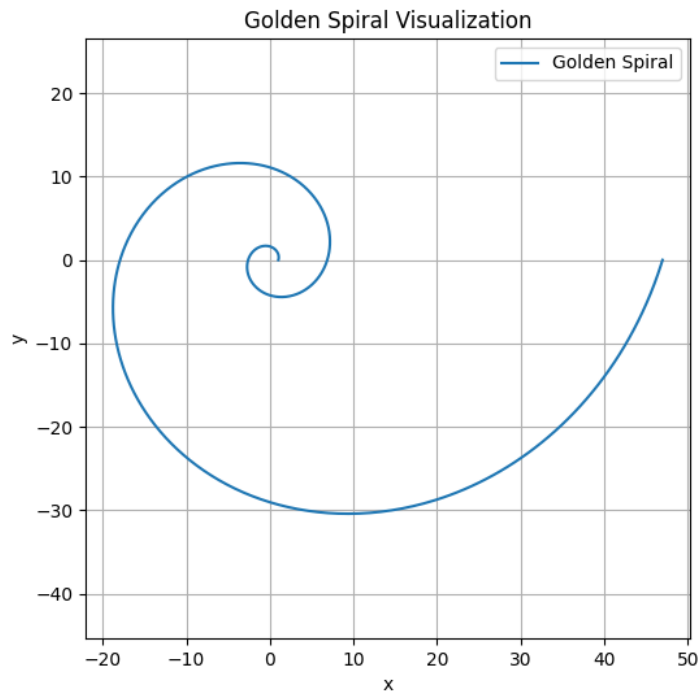


Figure 3.5: Visualization of the Golden Spiral

where r is the radius, φ is the angle, and a is a scale factor that controls the spread of the spiral. The inverse relationship between r and φ ensures that the spiral gradually widens while approaching an asymptotic line.

In the context of feature selection for wildfire prediction, the hyperbolic spiral's controlled expansion allows for a targeted exploration of the feature space. This expansion ensures the search progresses systematically without excessive jumps, improving the likelihood of identifying critical features for model accuracy.

Key Characteristics

- **Increasing Pitch Angle:** The hyperbolic spiral's pitch angle increases

with distance from the origin, supporting a decelerated yet comprehensive search space exploration.

- **Asymptotic Approach:** As the spiral expands, it approaches an asymptotic line, which limits extreme divergence and enables focused exploration.
- **Inverse Proportionality:** The inverse relationship between radius and angle allows the hyperbolic spiral to provide a balanced yet methodical expansion, making it suitable for optimization scenarios requiring gradual coverage.

Algorithm 20 Liver Cancer Algorithm (LCA) with Hyperbolic Spiral Update for Feature Selection

```
1: Step 1: Initialize Population
2: for each agent  $i$  in range (0, SearchAgents) do
3:   Generate a binary vector of length dimension for feature selection
4:   Generate opposite agents to enhance diversity and combine with initial
   agents
5: end for
6: Step 2: Calculate Initial Fitness
7: for each agent  $i$  do
8:   Select features based on agent  $i$ 's binary vector
9:   Evaluate fitness using cross-validation with XGBoost on the selected features
10:  Store fitness score in fit[ $i$ ]
11: end for
12: Identify the best initial solution, updating best_score and best_features
13: Step 3: Optimization Loop
14: for each iteration  $t$  in range (1, Max.iterations) do
15:   for each agent  $i$  do
```

```

16:      Exploration with Levy Flight: With probability 0.5, apply Levy
      flight to diversify the agent's position
17:      if random() < 0.5 then
18:          Generate a step using Levy flight and adjust the agent's position
19:      end if
20:      Hyperbolic Spiral Update: With probability 0.5, apply hyperbolic
      spiral update towards the best agent
21:      if random() < 0.5 then
22:          Set best_agent  $\leftarrow X[\text{argmax}(\text{fit})]$ 
23:          Detailed Logic of Hyperbolic Spiral Update:
24:          Compute the direction vector  $\text{direction} = \text{best\_agent} - X[i]$ 
25:          Calculate the Euclidean norm of the direction vector:
              
$$\text{norm} = \|\text{direction}\|$$

26:          if norm = 0 then
27:              Set norm = 1 to avoid division by zero
28:          end if
29:          Normalize the direction vector:
              
$$\text{unit\_direction} = \frac{\text{direction}}{\text{norm}}$$

30:          Set the angle increment  $\varphi$  and calculate the hyperbolic spiral step
      size using:
              
$$\text{step} = \frac{a}{\varphi} \cdot \text{unit\_direction}$$

31:          Update agent's position in the direction of the hyperbolic spiral step:
              
$$\text{new\_position} = X[i] + \text{step}$$

32:          Convert updated position to binary based on a dynamic threshold:
              
$$X[i] = (\text{new\_position} > \text{dynamic\_threshold}(t, \text{Max\_iterations})).\text{astype}(\text{int})$$


```

```

33:     end if
34:     Crossover and Mutation:
35:     Select a partner agent at random
36:     Apply crossover to generate a child:

        child  $\leftarrow$  crossover( $X[i]$ ,  $X[\text{partner\_idx}]$ )

37:     Apply mutation to the child:

        mutated_child  $\leftarrow$  mutation(child, mutation_rate)

38:     Calculate the fitness of the mutated child
39:     if mutated_child fitness  $\geq$  current agent fitness then
40:         Replace the current agent with the mutated child
41:     end if
42: end for
43: Update Best Solution:
44: if Current best fitness is greater than previous best then
45:     Update best_score and best_features
46: end if
47: end for
48: Step 4: Return Results
49: Output: The best solution best_features, best fitness score best_score, and
    fitness history over iterations

```

The Lituus Spiral

- **Lituus Spiral:** The Lituus spiral is an algebraic spiral characterized by an inverse square root relationship between the radius and the angle. This spiral

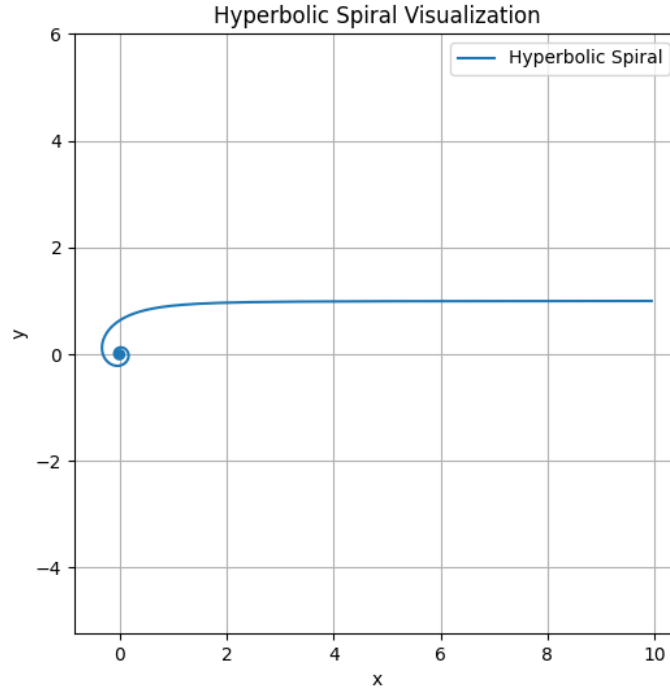


Figure 3.6: Visualization of the Hyperbolic Spiral

type is named after an ancient Roman trumpet called the "lituus," which Roger Cotes studied in 1722. The Lituus spiral has two symmetric branches, each approaching infinity asymptotically and winding towards the origin. The size of the spiral is controlled by the constant a , and each branch features an inflection point, creating a unique curvature pattern as it approaches the origin.

In polar coordinates, the Lituus spiral is represented by:

$$\rho = \frac{a}{\sqrt{\varphi}} \quad (3.32)$$

where ρ is the radius, φ is the angle in radians, and a is a constant that affects the scale of the spiral.

The Lituus spiral is advantageous for exploration tasks requiring controlled, inward-reaching searches. Its inverse relationship with the angle enables a targeted approach, which is particularly useful for gradually narrowing down a search area.

Key Characteristics

- **Inverse Square Root Relationship:** The radius is inversely proportional to the square root of the angle, resulting in a spiral that tightens near the origin.
- **Symmetric Branches:** Each branch is centrally symmetric around the origin, providing balanced coverage.
- **Inflection Points:** The spiral features inflection points at $(\varphi, \rho) = (\pm\frac{1}{2}, \sqrt{2a})$, where the curvature changes direction.

Algorithm 21 Liver Cancer Algorithm (LCA) with Detailed Lituus Spiral Update for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each agent i in range $(0, \text{SearchAgents})$ **do**
- 3: Generate a binary vector of length dimension for feature selection
- 4: Generate opposite agents to enhance diversity and combine with initial agents
- 5: **end for**
- 6: **Step 2: Calculate Initial Fitness**
- 7: **for** each agent i **do**
- 8: Select features based on agent i 's binary vector
- 9: Evaluate fitness using cross-validation with XGBoost on the selected features
- 10: Store fitness score in $\text{fit}[i]$

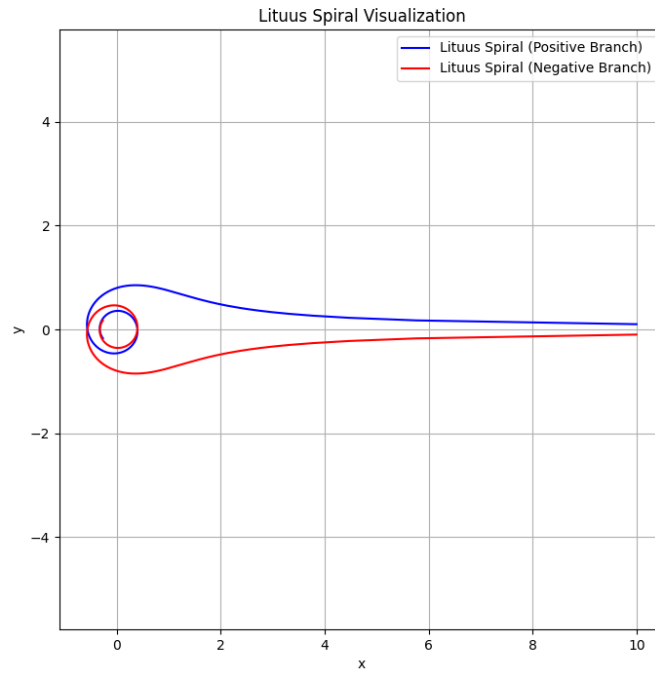


Figure 3.7: Visualization of the Lituus Spiral

```

11: end for
12: Identify the best initial solution, updating best_score and best_features
13: Step 3: Optimization Loop
14: for each iteration  $t$  in range (1, Max_iterations) do
15:   for each agent  $i$  do
16:     Exploration with Levy Flight: With probability 0.5, apply Levy
    flight to diversify the agent's position
17:     if random()  $\leq$  0.5 then
18:       Generate a step using Levy flight and adjust the agent's position
19:     end if
20:     Lituus Spiral Update: With probability 0.5, apply the Lituus spiral

```



```

    update towards the best agent
21:     if random() < 0.5 then
22:         Set best_agent  $\leftarrow X[\text{argmax}(\text{fit})]$ 
23:         Compute the direction vector  $\text{direction} = \text{best\_agent} - X[i]$ 
24:         Calculate the Euclidean norm of the direction vector:


$$\text{norm} = \|\text{direction}\|$$


25:         if norm = 0 then
26:             Set norm = 1 to avoid division by zero
27:         end if
28:         Normalize the direction vector:


$$\text{unit\_direction} = \frac{\text{direction}}{\text{norm}}$$


29:         Set the angle  $\varphi$  and calculate the Lituus spiral step size using:


$$\text{step} = \frac{a}{\sqrt{\varphi}} \cdot \text{unit\_direction}$$


30:         Update agent's position in the direction of the Lituus spiral step:


$$\text{new\_position} = X[i] + \text{step}$$


31:         Convert updated position to binary based on a dynamic threshold:


$$X[i] = (\text{new\_position} > 0.5).astype(int)$$


32:     end if
33:     Crossover and Mutation:
34:     Select a partner agent at random
35:     Apply crossover to generate a child:


$$\text{child} \leftarrow \text{crossover}(X[i], X[\text{partner\_idx}])$$


```

```

36:         Apply mutation to the child:

               mutated_child ← mutation(child, mutation_rate)

37:         Calculate the fitness of the mutated child
38:         if mutated_child fitness  $\geq$  current agent fitness then
39:             Replace the current agent with the mutated child
40:         end if
41:     end for
42:     Update Best Solution:
43:     if Current best fitness is greater than previous best then
44:         Update best_score and best_features
45:     end if
46: end for
47: Step 4: Return Results
48: Output: The best solution best_features, best fitness score best_score, and
               fitness history over iterations

```

The Logarithmic Spiral

- **Logarithmic Spiral:** The logarithmic spiral, also known as the equiangular or growth spiral, is a unique self-similar spiral where the distance between successive turns increases geometrically. This distinctive property results in a consistent or pitch angle between the spiral and any radial line, creating a curve that appears frequently in natural phenomena, such as shells and galaxies. Jacob Bernoulli, who extensively studied this spiral, famously called it *Spira mirabilis*, or "the marvellous spiral," due to its aesthetic appeal and mathematical properties.

In polar coordinates, the logarithmic spiral is represented by:

$$r = ae^{k\varphi} \quad (3.33)$$

where r is the radius, φ is the angle, a is a scaling constant, and k is a constant that defines the growth rate. The exponential relationship between r and φ ensures that the spiral's shape remains consistent as it expands.

In the context of feature selection for wildfire prediction, the logarithmic spiral's proportional growth provides a systematic way to explore the feature space. Its geometric progression prevents excessive clustering of candidate features, promoting a balanced search across the space to identify essential variables effectively.

Key Characteristics

- **Constant Pitch Angle:** The logarithmic spiral's pitch angle remains consistent, supporting systematic and even exploration across different scales in the search space.
- **Self-Similarity:** The self-similar nature of the logarithmic spiral ensures uniform coverage as the spiral progresses outward, making it ideal for optimization tasks that require balanced exploration and exploitation.
- **Exponential Growth:** The spiral's growth is geometric, increasing the distance between turns as it extends, which is advantageous for gradual exploration without crowding near the origin.

Algorithm 22 Liver Cancer Algorithm (LCA) with Detailed Logarithmic Spiral Update for Feature Selection

- 1: **Step 1: Initialize Population**
- 2: **for** each agent i in range $(0, \text{SearchAgents})$ **do**

```

3:   Generate a binary vector of length dimension for feature selection
4:   Generate opposite agents to enhance diversity and combine with initial
    agents
5: end for
6: Step 2: Calculate Initial Fitness
7: for each agent  $i$  do
8:   Select features based on agent  $i$ 's binary vector
9:   Evaluate fitness using cross-validation with XGBoost on the selected features
10:  Store fitness score in  $\text{fit}[i]$ 
11: end for
12: Identify the best initial solution, updating  $\text{best\_score}$  and  $\text{best\_features}$ 
13: Step 3: Optimization Loop
14: for each iteration  $t$  in range  $(1, \text{Max\_iterations})$  do
15:   for each agent  $i$  do
16:     Exploration with Levy Flight: With probability 0.5, apply Levy
      flight to diversify the agent's position
17:     if  $\text{random}() \leq 0.5$  then
18:       Generate a step using Levy flight and adjust the agent's position
19:     end if
20:     Logarithmic Spiral Update: With probability 0.5, apply logarithmic
      spiral update towards the best agent
21:     if  $\text{random}() \leq 0.5$  then
22:       Set  $\text{best\_agent} \leftarrow X[\text{argmax}(\text{fit})]$ 
23:       Detailed Logic of Logarithmic Spiral Update:
24:       Compute the direction vector  $\text{direction} = \text{best\_agent} - X[i]$ 
25:       Calculate the Euclidean norm of the direction vector:

```

$$\text{norm} = \|\text{direction}\|$$

```

26:         if norm = 0 then
27:             Set norm = 1 to avoid division by zero
28:         end if
29:         Normalize the direction vector:

$$\text{unit\_direction} = \frac{\text{direction}}{\text{norm}}$$

30:         Set the angular increment  $\theta$  and calculate the logarithmic spiral step
size:

$$\text{step} = ae^{k\theta} \cdot \text{unit\_direction}$$

31:         Update agent's position in the direction of the logarithmic spiral
step:

$$\text{new\_position} = X[i] + \text{step}$$

32:         Convert updated position to binary based on a dynamic threshold:

$$X[i] = (\text{new\_position} > \text{dynamic\_threshold}(t, \text{Max\_iterations})).\text{astype}(\text{int})$$

33:     end if
34:     Crossover and Mutation:
35:     Select a partner agent at random
36:     Apply crossover to generate a child:

$$\text{child} \leftarrow \text{crossover}(X[i], X[\text{partner\_idx}])$$

37:     Apply mutation to the child:

$$\text{mutated\_child} \leftarrow \text{mutation}(\text{child}, \text{mutation\_rate})$$

38:     Calculate the fitness of the mutated child
39:     if mutated_child fitness  $\geq$  current agent fitness then
40:         Replace the current agent with the mutated child

```

```

41:     end if
42: end for
43: Update Best Solution:
44: if Current best fitness is greater than previous best then
45:     Update best_score and best_features
46: end if
47: end for
48: Step 4: Return Results
49: Output: The best solution best_features, best fitness score best_score, and
    fitness history over iterations

```

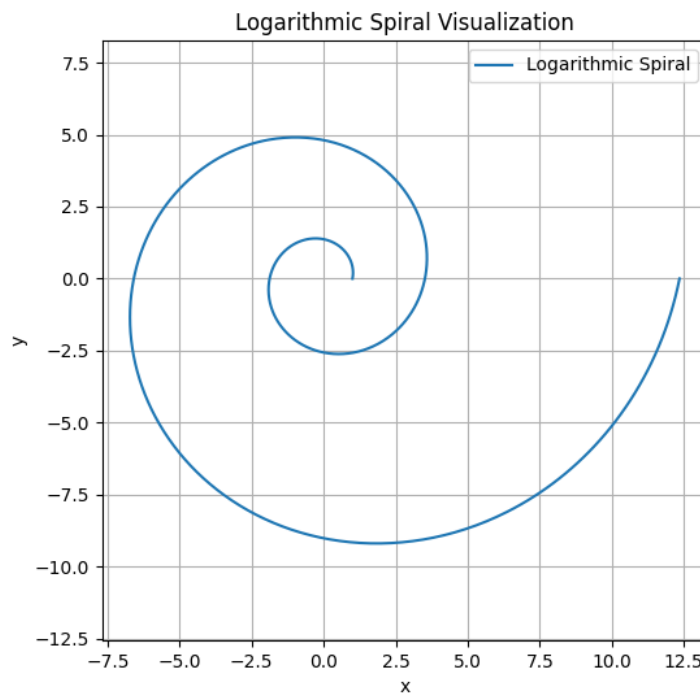


Figure 3.8: Visualization of the Logarithmic Spiral

In summary, each spiral update method Archimedean, Logarithmic, Hyperbolic,

or Lituus —offers unique advantages in navigating the solution space. These spirals provide diverse exploration-exploitation dynamics, enhancing our feature selection and optimization process.

To assess the impact of each spiral on algorithm performance, we define several evaluation metrics that allow us to measure effectiveness objectively across multiple runs. The following section outlines the key metrics used to quantify performance, stability, and feature selection capability, which form the basis for our results and comparative analysis.

3.6 Evaluation Metrics

To thoroughly evaluate the performance of our algorithm and its variations, we utilize several key metrics that provide a detailed assessment of effectiveness, stability, and feature selection capabilities. These metrics allow us to quantify the impact of the proposed methods across multiple executions, ensuring robustness and reliability.

3.6.1 Mean Fitness Function

The mean fitness function measures the algorithm’s average performance over multiple runs, indicating its effectiveness.

$$\text{Mean Fitness} = \frac{1}{n} \sum_{i=1}^n \text{fitness}(A_i) \quad (3.34)$$

Here, n is the number of runs, and $\text{fitness}(A_i)$ represents the fitness value of

Table 3.2: Comparison of Spiral Types for Feature Selection

Spiral Type	Growth Rate	Key Advantage	Best-Suited Scenario
Archimedean Spiral	Linear	Even, systematic spacing	Dense search spaces requiring uniform coverage
Euler Spiral	Linear Curvature	Smooth transitions, fine-tuned adjustments	High-dimensional spaces requiring gradual tuning
Fermat Spiral	Square Root	Gradual expansion, balanced search	Controlled exploration in sparse feature spaces
Golden Spiral	Exponential	Self-similarity, proportional growth	Complex datasets with multi-scale dependencies
Hyperbolic Spiral	Inverse Proportionality	Decelerated outward exploration	Focused searches in constrained feature spaces
Lituus Spiral	Inverse Square Root	Tight inward reach, symmetric branches	Gradual narrowing of high-impact regions
Logarithmic Spiral	Geometric	Constant pitch angle, balanced progression	Comprehensive exploration-exploitation balance

the solution in the i -th run.

3.6.2 Best Fitness Function

The best fitness function identifies the algorithm's optimal performance, highlighting its peak efficiency.

$$\text{Best Fitness} = \min_{i=1,\dots,n} \text{fitness}(A_i) \quad (3.35)$$

3.6.3 Worst Fitness Function

This metric records the worst-case performance, offering insights into the algorithm's consistency and identifying scenarios where improvements may be necessary.

$$\text{Worst Fitness} = \max_{i=1,\dots,n} \text{fitness}(A_i) \quad (3.36)$$

3.6.4 Standard Deviation

The standard deviation of fitness values measures the algorithm's stability and robustness across different runs.

$$\text{Standard Deviation} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\text{fitness}(A_i) - \text{Mean Fitness})^2} \quad (3.37)$$

3.6.5 Classification Accuracy (CA)

Classification accuracy evaluates the algorithm's ability to classify outcomes, averaging accuracy across all runs correctly.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.38)$$

Here, TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

3.6.6 Feature Selection Ratio (FSR)

The feature selection ratio quantifies the efficiency of the feature selection process, calculated as the proportion of selected features to the total features.

$$\text{FSR} = \frac{1}{n} \sum_{i=1}^n \frac{\sum A_i}{d} \quad (3.39)$$

Here, $\sum A_i$ represents the number of selected features in the i -th run, and d is the total number of features.

3.6.7 F-score

The F-score, the harmonic mean of precision and recall, evaluates the relevance and significance of the selected features.

$$\text{F-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.40)$$

3.6.8 Precision and Recall

Precision and recall provide further insights into the algorithm's performance. Precision measures the accuracy of optimistic predictions, while recall assesses the algorithm's ability to identify all relevant positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.41)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.42)$$

3.7 Statistical Tests

To objectively compare the standard Liver Cancer Algorithm (LCA) with its variations, we apply statistical tests assessing algorithm performance differences. These methods help establish whether observed improvements are statistically significant. The Wilcoxon Signed-Rank Test (Wilcoxon [1945]) and Friedman Test (Friedman [1937]) are employed to assess significant differences between algorithm performances.

3.7.1 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test is a nonparametric test used to compare paired samples and evaluate whether differences between two related datasets are statistically significant.

$$W = \sum_{i=1}^n \text{rank}(R_i) \cdot \text{sgn}(R_i) \quad (3.43)$$

Here, R_i represents the differences between paired observations, and $\text{sgn}(R_i)$ indicates the sign of the difference.

3.7.2 Friedman Test

The Friedman is a nonparametric test comparing multiple algorithms across several test scenarios. It identifies whether significant differences exist among the groups being tested.

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_{j=1}^k R_j^2 \right) - 3N(k+1) \quad (3.44)$$

In this equation:

- N : number of observations
- k : number of algorithms
- R_j : rank sum for the j -th algorithm

3.7.3 Algorithm Performance Analysis

This study systematically evaluates each algorithm’s performance using well-defined metrics to provide a comprehensive understanding of its strengths and limitations. Metrics such as Mean Fitness Function, Classification Accuracy, Recall, Precision, Feature Selection Ratio (FSR), and F-score are analyzed to capture each algorithm’s effectiveness, reliability, and feature selection quality.

Given the imbalanced nature of the dataset, we place particular emphasis on **Recall** and **Accuracy**. The recall is critical for assessing the algorithm’s ability to identify high-risk wildfire events accurately, minimizing false negatives, which could lead to unanticipated and severe consequences. Accuracy complements this by providing a broader measure of the algorithm’s overall predictive capability. To rank the algorithms fairly, we apply the Friedman and Wilcoxon Signed-Rank tests, which are robust non-parametric tools for comparing performance across multiple samples.

3.7.4 Regional Impact Analysis

We extend our evaluation to assess algorithm performance across specific Canadian provinces, recognizing that regional characteristics such as climate, vegetation density, and wildfire history can influence predictive accuracy. This regional analysis allows us to measure how well the algorithms adapt to diverse environmental conditions, highlighting computational or accuracy shifts that may reveal localized strengths or challenges.

To identify the most influential variables in each region, we analyze the top

features selected by each algorithm for specific provinces. We uncover regionally significant factors and universally impactful variables by ranking these features. This approach offers nuanced insights into the environmental factors critical for wildfire prediction, tailoring our findings to the unique conditions of each region.

3.8 Expected Outcomes and Limitations

Anticipated Findings: We expect some algorithms—particularly the refined Liver Cancer Algorithm (LCA) enhanced with spiral updates—to perform superiorly in certain climates or terrains. These variations may be particularly evident in regions with complex environmental dynamics, such as British Columbia or the Northwest Territories. Additionally, we anticipate identifying variables that show vital regional significance while recognizing universal predictors relevant across all provinces.

Potential Limitations: Class imbalance remains a significant challenge despite using cross-validation and stratified sampling to mitigate its effects. This imbalance could reduce recall for less frequent wildfire events, potentially leading to lower precision in identifying medium or low-risk zones. Furthermore, the large dataset imposes computational constraints, which may limit the depth of algorithmic comparisons or resolution in analyzing certain features.

3.9 Ethical and Practical Considerations

3.9.1 Ethical Data Use

This study adheres to strict ethical standards for data usage, mainly when working with publicly available environmental and geographical datasets such as FIRMS (NASA) and ERA5 (C3S). We follow guidelines from the data sources to ensure responsible data management practices to avoid misuse or misinterpretation. Privacy considerations are rigorously maintained, and results are disseminated with integrity, ensuring the findings are responsibly communicated.

3.9.2 Practical Implications

The findings of this research hold significant potential for wildfire management in Canada. By identifying critical predictive factors and optimizing algorithms for imbalanced datasets, our results can assist local agencies in managing wildfire risk more effectively. For example, predicting high-risk zones or critical periods could enable targeted firefighting efforts, faster response times, and risk mitigation. These insights aim to improve resource allocation and enhance resilience in wildfire management strategies across Canadian provinces.

3.10 Summary of the Study Framework

This study follows a comprehensive and methodical framework designed to achieve its goals rigorously and objectively. From initial feature selection to algorithmic

refinement, we applied various metaheuristic algorithms complemented by novel spiral updates to optimize predictive accuracy. Our evaluation framework emphasizes critical metrics for imbalanced datasets, such as Recall and accuracy, to ensure relevant and meaningful results. Statistical ranking tests and region-specific analyses further support our comparative approach.

Through structured phases, from data preprocessing to evaluation, the study underscores its commitment to advancing wildfire prediction tools and supporting resource management decisions. This robust framework sets the stage for the following Discussion section, where we will delve into a detailed analysis of our findings.

Chapter 4

Discussion

4.1 Overview of Study Objectives and Methodology

This study addresses the urgent need for accurate wildfire prediction tools by evaluating the effectiveness of various algorithms in different regions of Canada. We had two main objectives: (1) to assess the predictive performance of multiple algorithms—explicitly focusing on models enhanced with unique spiral updates—and (2) to examine how region-specific factors influence algorithm performance across provinces.

We employ a rigorous methodology to achieve these goals, including advanced feature selection techniques, comparative algorithm testing, and statistical classification using the Friedman and Wilcoxon signed-rank tests. We sourced data from NASA’s FIRMS and the Copernicus Climate Change Service’s ERA5, providing a comprehensive view of wildfire incidents across diverse environmental contexts.

We used cross-validation as a critical component during the training and testing phases to address the inherent class imbalance in wildfire data.

This approach allowed us to objectively compare each algorithm’s strengths and weaknesses while accounting for regional influences. The insights gained from this analysis aim to inform advancements in predictive modelling and practical applications in wildfire management across Canada.

4.1.1 Provincial Results

To provide a thorough analysis, we evaluated algorithm performance across eight provinces: Alberta, British Columbia (BC), Manitoba, Ontario, Northwest Territories (NWT), Quebec, Saskatchewan, and Yukon. This province-by-province approach highlights regional factors influencing prediction accuracy and provides insights into how well the algorithms adapt to varying environmental conditions. For each Canadian province, we focus primarily on **Recall** due to the imbalanced nature of wildfire data and the critical importance of identifying all high-risk wildfire cases. We also assess **Precision**, **Accuracy**, and **F1 Score** to provide a comprehensive view of each algorithm’s performance. This analysis helps tailor wildfire risk management strategies to the specific environmental conditions of each province. Initially, we rank the standard and spiral-enhanced algorithms using the Friedman test, which is particularly useful for identifying the most effective algorithms before and after spiral updates are applied. Subsequent detailed Wilcoxon signed-rank tests compare the original algorithms against their spiral-enhanced versions to ascertain any statistical performance improvements.

After establishing baseline algorithm rankings, we explore the impact of various spiral updates on the **Liver Cancer Algorithm (LCA)**. We reassess these

algorithms with another round of Friedman and Wilcoxon tests to evaluate the effectiveness of the enhancements. Results from these statistical tests are presented visually through comparative plots and tables, illustrating shifts in algorithm performance due to spiral enhancements. These visualizations and discussions highlight how specific algorithms and their enhancements improve the prediction and management of wildfires in different provincial contexts.

Algorithm Performance in Alberta

We evaluated the wildfire prediction algorithms in Alberta, focusing on their ability to identify high-risk wildfire cases. The results are summarized in Table 4.1.

Standard Algorithms: The **Particle Swarm Optimization (PSO)** algorithm achieved the highest average recall score of **79.79**, demonstrating exceptional effectiveness across all models in Alberta. Close behind, the **Golden Ratio Method (GRM)** performed robustly with an average recall of **78.32**, followed by the **CBO** optimization, which secured an average recall of **78.66**. These results underscore the capabilities of PSO, GRM, and CBO to effectively capture wildfire cases in Alberta.

Conversely, the **Atom Search Optimization (ASO)** algorithm had the lowest average recall score of **57.24**, indicating difficulties in adapting to Alberta's specific wildfire data characteristics. The **Liver Cancer Algorithm (LCA)** showed moderate performance with an average recall of **67.83**, which did not match the more successful algorithms.

A closer examination of the recall scores across individual models reveals interesting patterns. The **Random Forest** and **Gradient Boosting** models

consistently provided high recall scores for most algorithms, with PSO and CBO achieving **79.63** and **80.90**, respectively. This indicates a strong performance in capturing wildfire cases using these models.

Table 4.1: Algorithm Rankings for Alberta

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
PSO	79.63	83.88	73.07	85.94	76.43	10	2.00	1
CBO	80.90	81.56	72.00	85.47	73.40	19	3.80	2
GRM	79.30	83.20	71.98	86.80	70.30	21	4.20	3
GA	76.95	80.36	74.86	83.02	76.20	24	4.80	4
MRFO	80.04	80.08	72.89	74.93	75.52	25	5.00	5
EDO	78.58	81.12	70.99	84.70	73.85	26	5.20	6
BMO	78.13	80.19	70.14	83.97	75.35	27	5.40	7
EVO	81.06	80.07	71.80	77.13	74.72	29	5.80	8
WOA	78.37	79.50	70.68	85.31	69.63	30	6.00	9
LCA	72.61	71.58	70.90	60.54	63.52	45	9.00	10
EO	67.70	69.84	65.11	63.08	57.83	50	10.00	11
ASO	63.60	66.13	63.00	46.86	46.59	55	11.00	12

Spiral-Based Methods: Among the spiral-based methods, the **Euler Spiral** achieved the highest average Recall across most models, with **Random Forest** recording a recall of **80.72** and **Gradient Boosting** achieving **80.10**. This performance highlights the Euler Spiral’s ability to effectively balance search space exploration and exploitation, ensuring that essential feature subsets are not overlooked.

The **Hyperbolic Spiral** followed closely, delivering strong recall values, particularly **Random Forest** at **78.45** and **Gradient Boosting** at **78.39**. Its balance between diversification and intensification contributed to its competitive performance, achieving an **average recall of 74.45**.

Similarly, the **Lituus Spiral** demonstrated high recall values, including **Random Forest** at **77.95** and **K-Nearest Neighbor** at **76.44**, securing an **average recall of 74.38**. These results emphasize the Lituus Spiral’s capability to model complex patterns effectively.

The **Logarithmic Spiral**, with an **average recall of 73.36**, delivered consistent performance across models, such as **Random Forest** at **79.38** and **Gradient Boosting** at **78.41**. While slightly behind Hyperbolic and Lituus, its systematic exploration process maintained competitive performance.

Other methods, such as the **Fermat Spiral**, produced moderate results, achieving an **average recall of 72.62**, showing its ability to balance exploration and exploitation, although it lagged slightly behind the higher-performing spirals.

On the lower end, the **Liver Cancer Algorithm (LCA)** without spiral updates ranked consistently lower across most models, with a maximum recall of **72.61** achieved by **Random Forest** and an **average recall of 67.83**. This highlights the importance of integrating spiral updates to enhance LCA’s performance in addressing feature selection challenges.

Table 4.2: Spiral Algorithm Rankings for Alberta

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
Euler	80.72	80.10	71.80	76.05	75.50	8	1.60	1
Hyperbolic	78.45	78.39	73.14	69.73	72.53	13	2.60	2
Logarithmic	79.38	78.41	74.08	66.32	68.59	16	3.20	3
Lituus	77.95	77.19	76.44	68.77	71.55	17	3.40	4
Fermat	78.04	76.40	69.75	69.40	69.50	22	4.40	5
Golden	73.83	72.70	69.72	64.07	67.59	31	6.20	6
LCA	72.61	71.58	70.90	60.54	63.52	33	6.60	7
Archimedean	72.36	70.76	67.81	60.25	62.26	40	8.00	8

On the lower end, the **Liver Cancer Algorithm (LCA)** without spiral updates ranked consistently lower across most models, with a maximum recall of **72.61** achieved by **Random Forest**. This reinforces the impact of spiral updates in enhancing the LCA’s effectiveness.

Overall Algorithm Rankings: The Friedman test assessed the relative performance of all algorithms, including standard and spiral-based methods. Figure

4.1 visualizes the ranking of algorithms based on their average recall scores, where lower ranks indicate better performance.

The **Particle Swarm Optimization (PSO)** emerged as the best-performing algorithm with an average rank of **2.8**, followed by **CBO** at **4.6**, and **GRM** at **5.8**. The **Euler Spiral** performed notably well among the spiral methods, achieving an average rank of **6.6**. Other strong performers included **GA** at **6.0** and **MRFO** at **6.2**.

In contrast, algorithms such as **Archimedean Spiral** and **ASO** ranked lowest, with ranks of **17.2** and **19.0**, respectively. These results emphasize the effectiveness of top-ranked methods in feature selection and modeling, while highlighting areas for improvement in lower-performing approaches.

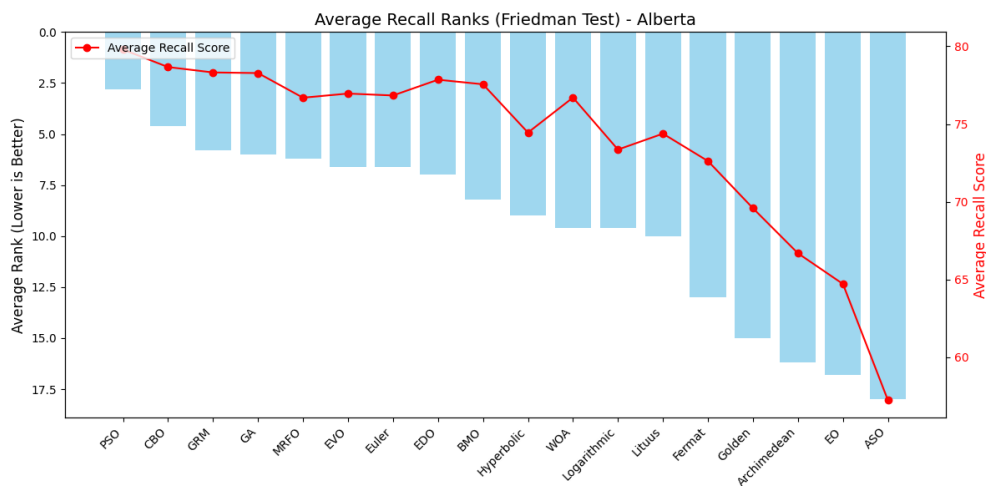


Figure 4.1: Average Recall Ranks (Friedman Test)

Algorithm Performance in British Columbia (BC)

The **Particle Swarm Optimization (PSO)** again led the performance in British Columbia with the highest average recall score of **75.60**, indicating its robustness across different regional datasets. The **Equilibrium Optimizer (EDO)** came close with an average recall of **74.99**, demonstrating effective wildfire prediction in this province.

The **Atom Search Optimization (ASO)** remained the least effective with the lowest average recall score of **39.61** in British Columbia, highlighting significant challenges in its application to wildfire data in this region. The **Manta ray foraging optimization (MRFO)** also underperformed with an average recall of **72.40**, showing some limitations in adapting to the specifics of British Columbia's wildfire scenarios.

In terms of individual model performances, the **Random Forest** and **Gradient Boosting** models again showed strong recall scores for top algorithms like PSO and EDO, with **78.71** and **78.95** respectively in British Columbia. This suggests that these models are particularly suited to leveraging the strengths of these algorithms for wildfire prediction in the region.

These insights highlight the varying effectiveness of different algorithms across provinces and models, suggesting that regional characteristics and data specifics significantly influence algorithm performance in wildfire prediction tasks.

Spiral-Based Enhancements: Among the spiral-based methods, the **LCA** achieved the highest average Recall, particularly excelling in **Random Forest** with a recall of **78.36** and **Gradient Boosting** at **78.24**. This performance highlights

Table 4.3: Algorithm Rankings for British Columbia

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
PSO	78.70	79.12	70.73	75.57	73.88	11	2.20	1
EDO	78.56	78.95	68.34	75.00	74.13	19	3.80	2
EVO	78.50	79.31	67.67	75.14	73.26	23	4.60	3
GA	78.18	79.13	68.08	75.84	73.48	23	4.60	3
BMO	78.23	79.16	69.97	74.91	72.56	26	5.20	5
GRM	78.27	77.88	70.05	71.82	74.29	26	5.20	5
WOA	78.22	78.94	68.83	75.35	72.80	28	5.60	7
LCA	78.36	78.24	67.35	74.53	73.44	33	6.60	8
CBO	77.78	78.11	67.86	73.52	72.98	40	8.00	9
MRFO	75.71	76.91	68.18	70.55	70.64	46	9.20	10
EO	71.81	72.38	64.64	64.13	65.64	55	11.00	11
ASO	38.92	38.67	40.27	40.44	39.77	60	12.00	12

Table 4.4: Spiral Algorithm Rankings for British Columbia

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
LCA	78.36	78.24	67.35	74.53	73.44	10	2.00	1
Archimedean	78.08	78.26	70.28	72.23	71.12	12	2.40	2
Hyperbolic	76.30	76.04	70.54	73.28	72.33	14	2.80	3
Golden	76.85	76.53	68.33	70.06	69.98	20	4.00	4
Euler	75.39	74.58	68.41	71.96	72.48	20	4.00	4
Litmus	73.31	74.84	65.89	64.86	66.45	30	6.00	6
Logarithmic	72.76	72.58	63.15	65.30	65.31	34	6.80	7
Fermat	56.42	55.88	50.92	50.05	48.05	40	8.00	8

LCA’s strong feature selection capabilities, surpassing other spirals.

The **Archimedean Spiral** followed closely, demonstrating solid recall values with **Random Forest** at **78.08** and **Gradient Boosting** at **78.26**. Its ability to balance exploration and exploitation contributed to its competitive performance.

The **Golden Spiral** also showed promise, maintaining consistent recall scores across models, including **76.85** in **Random Forest** and **76.53** in **Gradient Boosting**.

Other spirals, such as **Hyperbolic** and **Euler**, demonstrated moderate improvements with average recalls of **73.30** and **72.16**, respectively. These results underscore the adaptability of spiral-based approaches in enhancing predictive performance.

On the lower end, the **Fermat Spiral** ranked lowest, with an average recall of **52.66**, highlighting areas for potential optimization in feature selection strategies.

Statistical Validation: The Friedman test results validated the performance differences among the algorithms, yielding significant statistics (**Statistic: 69.25, P-value: 0.00000** and **Statistic: 71.81, P-value: 0.00000**). Rankings further highlighted the dominance of the **LCA** and **Archimedean Spirals** over other methods. The **LCA Standard** ranked 7.6, whereas the **Archimedean Spiral** ranked 8.2, followed by the **Hyperbolic Spiral** at 9.4. Conversely, lower-ranked methods such as **Logarithmic Spiral** (16.2) and **Fermat Spiral** (18.0) demonstrated relatively weaker performance, emphasizing the superior adaptability of top-performing algorithms in modeling complex relationships and optimizing feature selection.

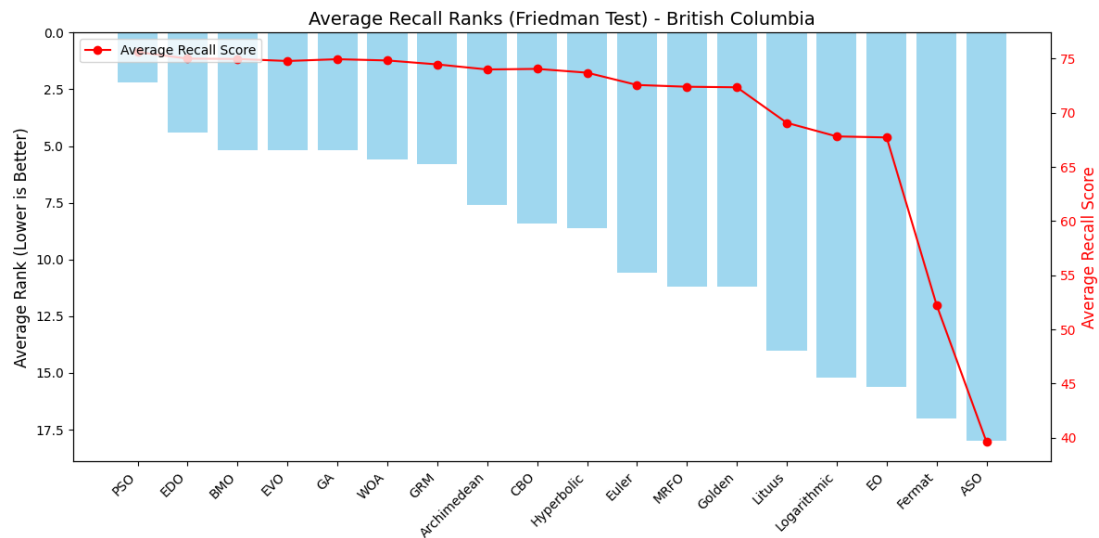


Figure 4.2: Visualization of average recall rankings among algorithms in BC (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA.

Overall Algorithm Rankings: The Friedman test assessed the relative performance of all algorithms, including standard and spiral-based methods. Figure 4.1 visualizes the ranking of algorithms based on their average recall scores, where lower ranks indicate better performance.

The **Particle Swarm Optimization (PSO)** emerged as the best-performing algorithm with an average rank of **2.2**, followed by **EDO** at **4.4** and **EVO** at **5.4**. Among spiral-based methods, the **Euler Spiral** achieved an average rank of **11.4**, outperforming several algorithms. However, **Archimedean** and **Hyperbolic** also performed competitively, ranking **8.2** and **9.4**, respectively. On the lower end, methods like **Logarithmic** (16.2) and **Fermat** (18.0) demonstrated weaker performance, emphasizing the dominance of top-ranked algorithms in enhancing feature selection and predictive accuracy.

Algorithm Performance in Manitoba

The results are summarized in Table 4.5. The **Genetic Algorithm (GA)** showcased the best average recall in Manitoba, leading with an average score of **69.63**. It demonstrated robust performance across all models, affirming its capability in handling diverse data scenarios effectively. Following closely were the **Equilibrium Optimizer (EVO)** and the **Exponential Distribution Optimizer (EDO)**, with average recalls of **69.53** and **69.39** respectively, highlighting their efficiency in optimizing prediction models under varying conditions.

On the other hand, the **Atom Search Optimization (ASO)** lagged significantly behind other algorithms with the lowest average recall of **44.34**, suggesting that ASO might be less suited to the specific challenges presented by the Manitoba dataset. The algorithms like **Equilibrium Optimizer** and **CBO** performed well,

especially in models such as **Random Forest** and **Logistic Regression**, indicating a strong adaptability to the data characteristics in Manitoba. .

Table 4.5: Algorithm Rankings for Manitoba

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
EVO	72.30	73.35	63.29	73.53	65.17	18	3.60	1
CBO	71.22	70.93	63.82	72.83	67.88	19	3.80	2
LCA	72.47	70.98	62.96	71.04	67.20	20	4.00	3
GA	71.11	72.30	65.72	72.75	66.29	22	4.40	4
MRFO	72.42	70.61	62.59	72.24	66.36	24	4.80	5
EDO	71.38	70.55	61.89	75.48	67.65	25	5.00	6
PSO	72.09	70.31	64.59	68.55	67.04	29	5.80	7
BMO	71.92	71.66	62.11	69.77	64.49	34	6.80	8
WOA	71.19	70.51	60.30	71.20	66.02	39	7.80	9
GRM	67.67	68.70	62.59	69.01	64.71	44	8.80	10
EO	63.79	61.43	56.89	64.54	60.43	55	11.00	11
ASO	43.65	42.94	46.07	45.32	43.73	60	12.00	12

Spiral-Based Enhancements: Among the spiral-based methods evaluated in Manitoba, the **Liver Cancer Algorithm (LCA)** emerged as the top performer, demonstrating its strong feature selection capabilities. It particularly excelled in **Random Forest** with a recall of **72.47** and **Gradient Boosting** with a recall of **70.98**. These results underscore the LCA’s ability to effectively model complex patterns in wildfire prediction.

Following closely was the **Archimedean Spiral**, which showed solid performance across models, with recall scores of **69.32** in **Random Forest** and **69.27** in **Gradient Boosting**. Its balance between exploration and exploitation contributed significantly to its competitive performance.

The **Hyperbolic Spiral** also demonstrated robust performance, particularly with recall values of **65.58** in **Random Forest** and **66.46** in **Gradient Boosting**. This method’s blend of diversification and intensification strategies has proven effective in navigating the challenges of wildfire prediction.

Meanwhile, the **Euler Spiral** and **Lituus Spiral** achieved moderate improve-

ments, with average recalls of **61.78** and **59.41** respectively. These spirals are notable for their methodical approach to modeling, yet they show potential for further optimization.

The **Logarithmic Spiral** provided consistent results across various models but fell slightly behind the leading spirals with an average recall of **60.61**. Its systematic exploration process maintains a competitive edge, although there is room for enhancement.

On the lower end, the **Fermat Spiral** and **Golden Spiral** demonstrated areas for potential improvement, with the lowest recalls of **54.76** and **56.41** respectively. These results highlight the need for further refinement in their algorithms to better tackle the complexities of wildfire prediction in Manitoba.

Table 4.6: Spiral Algorithm Rankings for Manitoba

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
LCA	72.47	70.98	62.96	71.04	67.20	6	1.20	1
Archimedean	69.32	69.27	65.12	68.21	66.19	9	1.80	2
Hyperbolic	65.58	66.46	60.46	67.25	66.11	15	3.00	3
Euler	62.71	62.12	59.35	64.97	59.73	25	5.00	4
Logarithmic	64.77	62.65	54.90	65.06	55.64	26	5.20	5
Litius	61.80	62.14	55.28	65.22	54.60	27	5.40	6
Fermat	57.93	57.27	55.25	51.18	52.14	36	7.20	7
Golden	57.79	57.12	55.20	55.20	54.76	36	7.20	7

Statistical Validation: The Friedman test confirmed significant differences in algorithm performance, with a test statistic of **79.62** and a p-value of **9.99e-10**, indicating strong evidence against the null hypothesis of equal performance. Figure [4.3](#) visualizes the rankings of all algorithms.

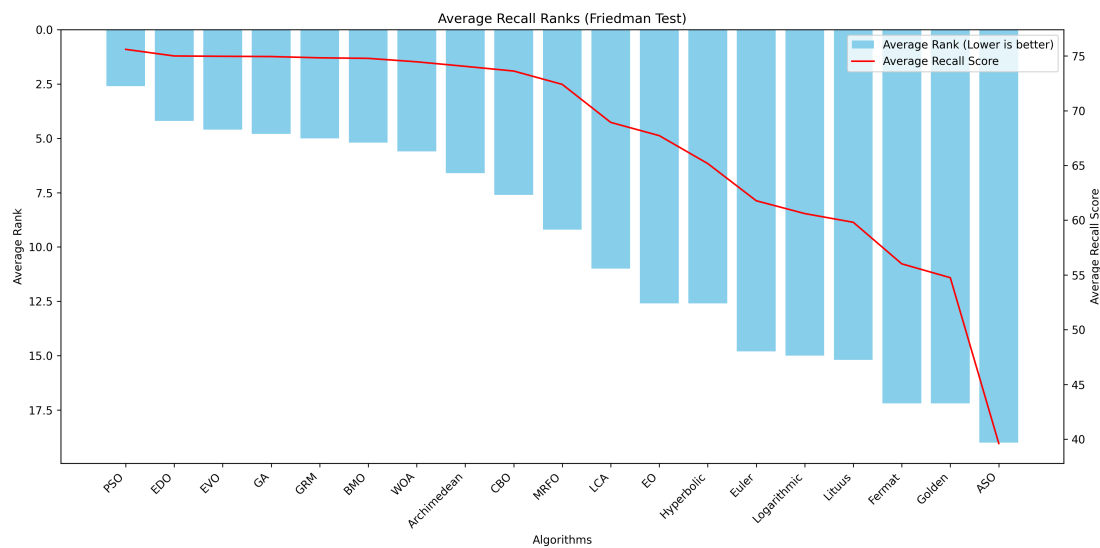


Figure 4.3: Visualization of average recall rankings among algorithms in Manitoba (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA.

Algorithm Performance in Northwest Territories (NWT)

In the Northwest Territories, the **Manta Ray Foraging Optimization (MRFO)** stood out with the highest average recall of **75.49**, demonstrating its efficiency in optimizing model parameters under varying conditions. The **Compound Binary Optimization (CBO)** also performed well, achieving a similar high average recall of **74.81**, particularly excelling in logistic regression and support vector machine models.

Conversely, the **Atom Search Optimization (ASO)** and **Equilibrium Optimizer (EO)** showed the least effective performance with the lowest average recalls of **51.99** and **51.46** respectively. These results suggest that these algorithms may require further tuning or adaptation to better suit the data peculiarities in NWT.

Table 4.7: Algorithm Rankings for Northwest Territories (NWT)

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
MRFO	76.32	75.90	71.69	80.99	72.54	12	2.40	1
BMO	75.16	74.83	71.19	79.29	73.35	21	4.20	2
CBO	76.14	74.73	72.00	79.67	71.50	21	4.20	2
PSO	74.96	75.91	69.91	79.98	71.84	21	4.20	2
EVO	74.90	76.93	69.23	79.33	71.80	25	5.00	5
GA	73.68	76.17	67.19	80.61	72.37	26	5.20	6
WOA	75.30	77.45	69.13	78.01	70.25	26	5.20	6
EDO	74.38	77.15	67.02	78.78	70.04	33	6.60	8
GRM	73.36	73.36	69.65	74.87	68.44	43	8.60	9
LCA	73.91	72.79	64.82	76.73	67.94	47	9.40	10
EO	51.93	53.03	56.65	44.05	51.62	57	11.40	11
ASO	52.39	52.79	52.84	50.80	51.15	58	11.60	12

Spiral Algorithm Performance in Northwest Territories (NWT) In the Northwest Territories, the effectiveness of spiral-based algorithms was thoroughly evaluated for wildfire prediction. The outcomes are detailed in Table 4.7.

Among the spiral-based methods, the **Archimedean Spiral** stood out as the top performer, achieving the highest average recall scores across most models. Specifically, it excelled in **Logistic Regression** with a recall of **77.65** and demonstrated consistent performance in **Random Forest** and **Gradient Boosting** with recalls of **75.87** and **76.10** respectively. This reflects its robust ability to navigate the feature space effectively, balancing both exploration and exploitation.

Following closely was the **Golden Spiral**, which showed strong recall values, particularly in **Logistic Regression** with a recall of **75.84** and **Gradient Boosting** at **75.08**. This method's consistent performance across models indicates its potential in handling diverse and complex data structures inherent to wildfire prediction.

The **Liver Cancer Algorithm (LCA)** also demonstrated commendable performance, with an average recall of **71.64**. It showed a particularly strong recall in **Logistic Regression** at **76.73**, proving its efficacy in specific scenarios despite

a slightly lower performance in **K-Nearest Neighbor**.

Other methods such as the **Logarithmic Spiral** and **Hyperbolic Spiral** delivered moderate results with average recalls of **68.79** and **66.26**, respectively. These algorithms, while effective to a degree, indicate the need for further tuning to optimize their performance in the challenging environments of the Northwest Territories.

The **Fermat Spiral** and **Euler Spiral** showed some limitations in achieving higher recall values, with the **Euler Spiral** in particular finding challenges in **Logistic Regression** and **Support Vector Machine** with lower recall scores of **55.98** and **62.10** respectively.

At the lower end of the performance spectrum, the **Lituus Spiral** recorded the lowest overall recall of **64.29**, struggling particularly in **Logistic Regression** and **K-Nearest Neighbor** with recalls of **58.36** and **62.72**. This suggests that while the Lituus Spiral may have potential, it requires significant adjustments to better address the specific demands of wildfire prediction in this region.

Overall, the evaluation of spiral-based algorithms in the Northwest Territories illustrates a varied landscape of effectiveness, with some algorithms showing promise in specific models while others require enhancements to reach their full potential in wildfire prediction.

Table 4.8: Spiral Algorithm Rankings for Northwest Territories (NWT)

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
Archimedean	75.87	76.10	71.13	77.65	72.05	7	1.40	1
Golden	74.45	75.08	71.27	75.84	73.12	9	1.80	2
LCA	73.91	72.79	64.82	76.73	67.94	17	3.40	3
Logarithmic	70.67	71.03	67.11	68.11	67.01	20	4.00	4
Fermat	69.80	67.63	65.59	66.20	65.21	30	6.00	5
Euler	70.16	69.58	67.79	55.98	62.10	30	6.00	5
Hyperbolic	66.85	68.33	64.09	66.90	65.10	33	6.60	7
Lituus	68.98	69.94	62.72	58.36	63.44	34	6.80	8

Statistical Validation: The Friedman test confirmed statistically significant differences in algorithm performance, with a test statistic of **71.49** and a p-value of **2.52e-08**. Figure 4.4 illustrates the ranking distribution, highlighting the superior performance of spiral-based enhancements over the standard LCA.

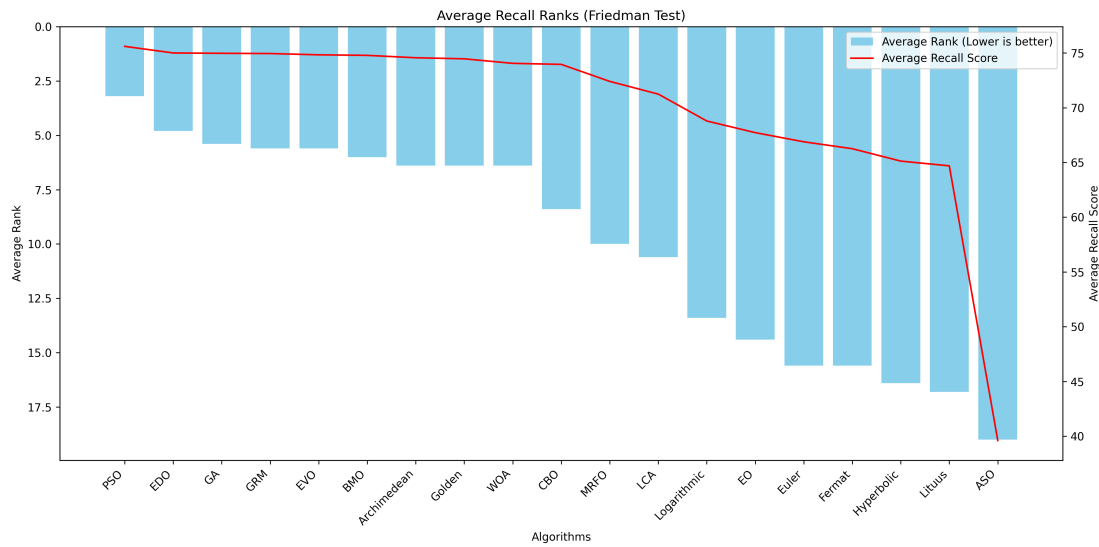


Figure 4.4: Visualization of average recall rankings among algorithms in Northwest Territories (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA.

Algorithm Performance in Ontario

In Ontario, the **Equilibrium Optimizer (EVO)** emerged as the top-performing algorithm with the highest average recall of **69.73**, showcasing its effectiveness across diverse models, particularly in the **K-Nearest Neighbor** model where it achieved exceptionally high recall. The **Manta Ray Foraging Optimization (MRFO)** followed closely with an average recall of **68.80**, indicating its robust capability in handling complex datasets.

The **Atom Search Optimization (ASO)**, however, recorded the lowest performance with an average recall of **44.67**, reflecting challenges in adapting to the specific characteristics of the Ontario dataset. The algorithms such as **Exponential Distribution Optimizer (EDO)** and **Genetic Algorithm (GA)** also showed good performance, particularly in logistic regression, underlining their strong adaptability.

Table 4.9: Algorithm Average Recall and Rankings for Ontario

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Average Recall	Ranking
ASO	46.40	46.24	35.45	47.85	47.40	44.67	12
BMO	67.99	66.74	62.04	71.08	64.83	66.53	5
CBO	63.63	66.15	61.23	73.40	62.15	65.31	6
EVO	64.47	65.75	82.43	74.24	61.77	69.73	1
EO	54.82	52.64	47.09	51.44	49.79	51.16	11
EDO	68.16	68.13	63.71	72.74	65.55	67.66	3
GA	66.41	66.48	65.22	73.70	64.16	67.19	4
GRM	64.55	65.66	52.77	69.66	63.65	63.26	9
LCA	58.98	57.09	51.57	66.53	60.07	58.85	10
MRFO	68.16	68.15	66.29	72.44	68.94	68.80	2
PSO	64.42	63.30	60.01	71.51	64.45	64.74	7
WOA	64.20	63.69	56.83	71.45	62.41	63.72	8

Spiral Algorithm Performance in Ontario In Ontario, the effectiveness of spiral-based algorithms in wildfire prediction was systematically evaluated, with summarized results presented in Table 4.10.

The **Hyperbolic Spiral** emerged as the leader among the spiral algorithms, showcasing the highest average recall values. It demonstrated superior performance particularly in **Random Forest** and **Gradient Boosting** with recall scores of **70.21** and **69.34** respectively. This highlights its efficient balance of exploration and exploitation tactics in complex prediction scenarios.

Closely following was the **Archimedean Spiral**, which secured the second position. It recorded strong recall scores, especially in **Random Forest** and **Logistic Regression**, with values of **68.09** and **64.62**. Its methodical approach

to navigating the feature space has proven effective, albeit slightly less so than the Hyperbolic Spiral.

The **Fermat Spiral** also demonstrated commendable results, with an average recall of **63.20**, performing particularly well in **Logistic Regression** with a recall of **62.91**. Its consistent performance across different models underscores its potential in handling diverse data environments.

Other algorithms such as the **Logarithmic Spiral** and **Lituus Spiral** showed moderate effectiveness with average recalls of **62.33** and **61.52** respectively. These results suggest that while capable, they may require further optimization to enhance their predictive accuracy.

The **Euler Spiral** and **Golden Spiral** ranked lower, indicating room for improvement in their feature selection processes. They managed average recalls of **60.53** and **60.33**, highlighting the challenges they face in adapting to the complex wildfire prediction landscape in Ontario.

At the lower end of the performance spectrum, the **Liver Cancer Algorithm (LCA)** exhibited the least effective performance, with an average recall of **58.05**. Despite showing strong performance in **Logistic Regression**, its overall lower scores reflect significant limitations in its current configuration for the Ontario dataset.

Statistical Validation: The Friedman test confirmed significant differences in algorithm performance, with a test statistic of **61.98** and a p-value of **9.74e-07**, indicating strong evidence against the null hypothesis of equal performance. Figure [4.5](#) illustrates the ranking distribution, showcasing the superior performance of spiral-based enhancements over the standard LCA and other algorithms.

Table 4.10: Algorithm Rankings for Ontario

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
MRFO	68.16	68.15	66.29	72.44	68.94	10	2.00	1
EDO	68.16	68.13	63.71	72.74	65.55	13	2.60	2
GA	66.41	66.48	65.22	73.70	64.16	18	3.60	3
BMO	67.99	66.74	62.04	71.08	64.83	22	4.40	4
EVO	64.47	65.75	82.43	74.24	61.77	23	4.60	5
CBO	63.63	66.15	61.23	73.40	62.15	31	6.20	6
PSO	64.42	63.30	60.01	71.51	64.45	33	6.60	7
GRM	64.55	65.66	52.77	69.66	63.65	36	7.20	8
WOA	64.20	63.69	56.83	71.45	62.41	38	7.60	9
LCA	58.98	57.09	51.57	66.53	60.07	50	10.00	10
EO	54.82	52.64	47.09	51.44	49.79	55	11.00	11
ASO	46.40	46.24	35.45	47.85	47.40	60	12.00	12

Algorithm Performance in Quebec

In Quebec, the **Particle Swarm Optimization (PSO)** algorithm demonstrated the highest overall effectiveness with an average recall of **78.69**, showcasing its strong performance across all models, particularly excelling in logistic regression with an impressive recall of **88.40**. This highlights PSO's capability to adapt well to the diverse modeling challenges presented by the Quebec dataset.

The **Equilibrium Optimizer (EVO)** also performed admirably, securing the second-highest average recall of **78.30**. It was particularly effective in handling the K-Nearest Neighbor model, indicating its robustness in managing complex, non-linear data structures.

On the other hand, the **Atom Search Optimization (ASO)** struggled in Quebec, recording the lowest average recall of **54.21**. This suggests potential difficulties in parameter tuning or in adapting to the specific characteristics of the data used in Quebec.

Spiral-Based Enhancements in Quebec: In Quebec, the evaluation of spiral-based algorithms revealed significant variations in their effectiveness for predicting

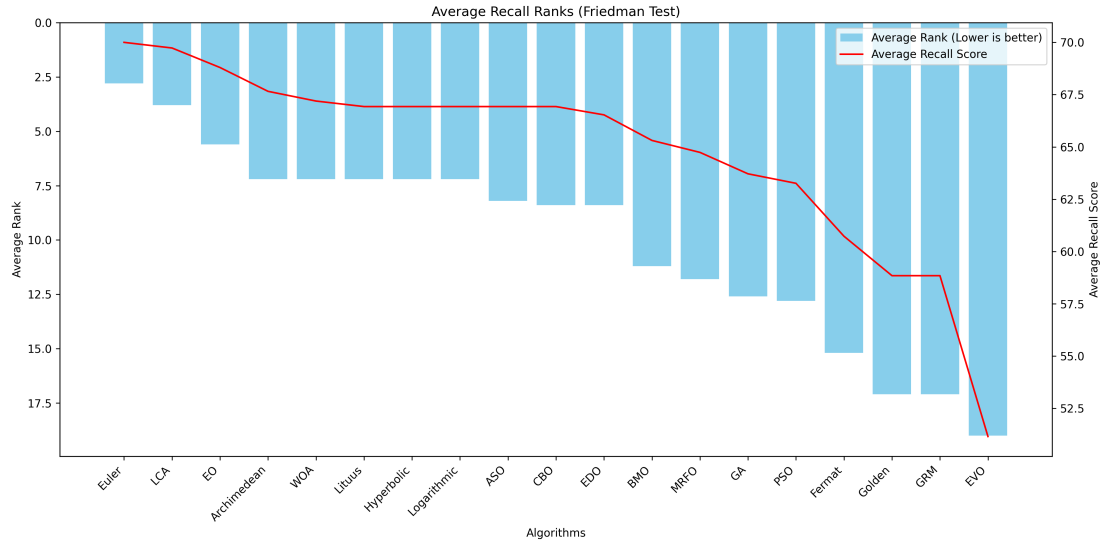


Figure 4.5: Visualization of average recall rankings among algorithms in Ontario (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA.

wildfires, as detailed in Table 4.11.

The **Archimedean Spiral** led the rankings, achieving the highest average recall. It showcased exceptional performance in **Logistic Regression** with a recall of **86.12**, and in **Gradient Boosting**, where it achieved a recall of **81.64**. This algorithm’s ability to adeptly navigate complex data landscapes is evident from its strong performance across diverse models.

Following closely, the **Logarithmic Spiral** demonstrated robust recall scores, particularly in **Gradient Boosting** and **Logistic Regression** with recalls of **81.56** and **79.39**, respectively. This indicates its effective use of logarithmic progression to optimize feature selection and model training processes.

The **Euler Spiral** also performed well, with notable strength in **Gradient Boosting** where it recorded a recall of **83.38**. Its unique approach to balancing

Table 4.11: Algorithm Rankings for Quebec

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
PSO	78.97	81.38	73.57	88.40	71.13	14	2.80	1
GRM	79.30	83.20	71.98	86.80	70.30	15	3.00	2
EDO	78.58	81.12	70.99	84.70	73.85	21	4.20	3
EVO	82.07	82.30	69.60	84.59	72.93	22	4.40	4
CBO	79.26	84.39	68.00	87.06	69.40	25	5.00	5
BMO	78.13	80.19	70.14	70.14	83.97	29	5.80	6
WOA	78.37	79.50	70.68	85.31	69.63	30	6.00	7
MRFO	76.33	75.77	72.15	83.53	70.06	34	6.80	8
GA	77.10	76.44	70.04	80.01	70.29	38	7.60	9
EO	67.70	69.84	70.14	63.08	57.83	49	9.80	10
LCA	68.41	67.24	66.30	68.22	61.49	52	10.40	11
ASO	54.59	54.56	53.80	54.13	53.99	60	12.00	12

computational efficiency with exploratory data analysis has contributed to its competitive average recall of **76.40**.

Not to be overlooked, the **Hyperbolic Spiral** provided solid results with a particularly strong performance in **Random Forest** achieving a recall of **79.15**. Its method of focusing on diversification in the feature space aligns well with the dynamic and variable-intensive nature of wildfire prediction.

On the other end, the **Fermat Spiral** and the **Golden Spiral** demonstrated moderate effectiveness with average recalls of **72.58** and **72.77**, respectively. While they showed potential in specific models, their overall performance suggests room for optimization to fully harness their algorithms' capabilities in the Quebec environment.

The **Lituus Spiral** and the **Liver Cancer Algorithm (LCA)** encountered some challenges, with the LCA showing the least adaptability among the evaluated spirals, holding the lowest average recall of **66.33**. This underlines the necessity for potential algorithmic adjustments or enhanced feature engineering to improve its application in this specific regional context.

Table 4.12: Spiral Algorithm Rankings for Quebec

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
Archimedean	77.78	81.64	75.78	86.12	71.30	9	1.80	1
Logarithmic	77.56	81.56	74.46	79.39	72.43	14	2.80	2
Euler	77.45	83.38	70.96	81.86	70.35	14	2.80	2
Hyperbolic	79.15	75.74	69.45	78.63	72.62	15	3.00	4
Fermat	75.05	71.26	65.92	80.70	69.96	28	5.60	5
Golden	74.77	74.71	68.76	72.38	69.24	29	5.80	6
Litius	72.44	72.10	68.42	71.04	69.91	32	6.40	7
LCA	68.41	67.24	66.30	68.22	61.49	39	7.80	8

Statistical Validation: The Friedman test confirmed significant differences in algorithm performance, with a test statistic of **60.56** and a p-value of **1.66e-06**, indicating strong evidence against the null hypothesis of equal performance. Figure 4.6 illustrates the ranking distribution, highlighting the superior performance of spiral-based enhancements over the standard LCA.

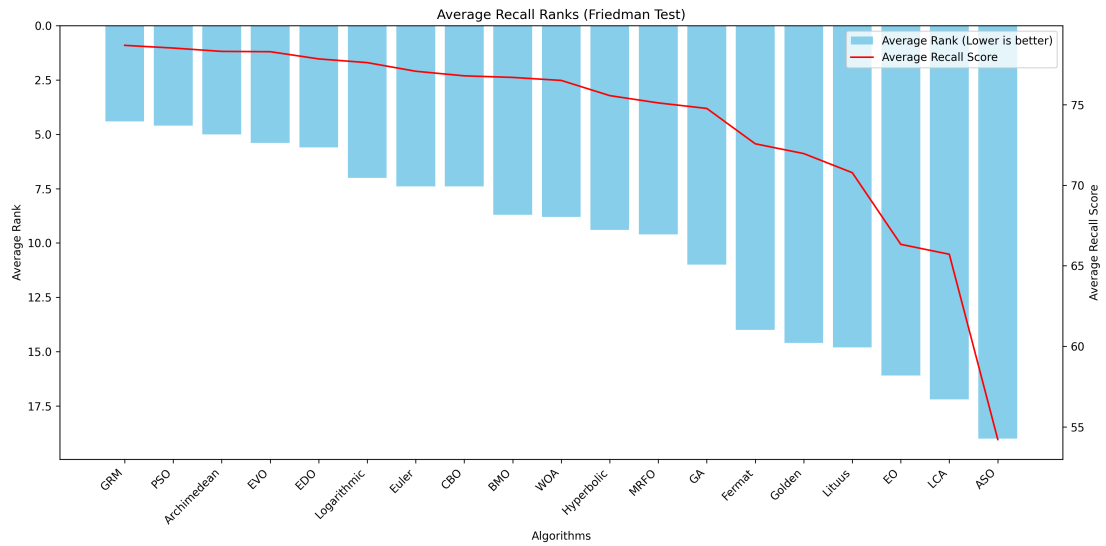


Figure 4.6: Visualization of average recall rankings among algorithms in Quebec (Friedman Test results). Lower ranks indicate better performance, with several spiral-enhanced algorithms outperforming the LCA.

Algorithm Performance in Saskatchewan

In Saskatchewan, the **Genetic Algorithm (GA)** emerged as the top performer with the highest average recall of **78.78**, indicating its strong adaptability and effectiveness across various models. This was particularly notable in the Support Vector Machine and Logistic Regression models, where it achieved recalls of **78.20** and **77.91** respectively.

The **Particle Swarm Optimization (PSO)** and the **Whale Optimization Algorithm (WOA)** also showed robust performances with average recalls of **78.33** and **78.01**, respectively, securing them positions among the top three algorithms. These results underscore the capability of PSO and WOA to efficiently navigate the predictive modeling landscape in Saskatchewan.

Conversely, the **Atom Search Optimization (ASO)** had the lowest average recall of **47.26** in Saskatchewan, similar to its performance in Quebec. This consistent underperformance across different provinces may necessitate a reevaluation of its application or further optimization to better suit the regional data specifics.

Table 4.13: Algorithm Rankings for Saskatchewan

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
GA	80.62	81.10	76.07	77.91	78.20	10	2.00	1
PSO	80.83	80.78	75.01	78.36	76.67	11	2.20	2
WOA	80.87	80.93	73.55	78.13	76.55	11	2.20	2
BMO	80.30	80.34	73.22	77.90	76.06	24	4.80	4
EVO	80.80	80.27	71.74	77.49	76.23	26	5.20	5
CBO	80.24	80.90	72.07	74.52	74.47	30	6.00	6
GRM	80.14	80.55	72.40	73.62	71.80	35	7.00	7
EDO	79.92	79.91	71.51	76.45	74.32	39	7.80	8
MRFO	79.07	80.14	71.46	75.37	74.70	39	7.80	8
EO	76.97	77.12	69.37	68.76	71.36	50	10.00	10
LCA	75.92	75.13	62.10	68.51	67.64	55	11.00	11
ASO	47.40	50.46	51.79	44.86	41.81	60	12.00	12

Spiral-Based Enhancements: The evaluation of spiral-based algorithms for wildfire prediction in Saskatchewan has highlighted varying levels of effectiveness across different models, as summarized in Table 4.13.

Leading the pack was the **Archimedean Spiral**, which showcased superior performance across all models. It achieved impressive recall scores in **Random Forest** and **Gradient Boosting** with values of **81.02** and **80.28**, respectively, culminating in the highest average recall of **77.38**. This spiral's ability to effectively explore and exploit the feature space makes it particularly suited to the predictive challenges in Saskatchewan.

The **Lituus Spiral** and the **Euler Spiral** both shared the second rank with an average recall of **77.20**. These algorithms demonstrated strong performances, particularly in **Logistic Regression** and **Support Vector Machine**, where they matched each other with recalls of **77.39** and **76.08**, indicating their robustness in handling diverse data structures.

The **Fermat Spiral** secured the third rank with a solid average recall of **75.66**. While it trailed slightly behind the leaders, it still showed strong potential in models like **Random Forest** and **Gradient Boosting** with recalls of **79.14** and **78.75**, suggesting that with slight optimizations, it could compete closely with the top performers.

Subsequently, the **Logarithmic Spiral** and the **Golden Spiral** ranked fourth and fifth, with average recalls of **75.19** and **74.31**, respectively. These spirals, while effective, displayed a need for enhancements to elevate their performance to the top tier.

On the lower end, the **Hyperbolic Spiral** and the **Liver Cancer Algorithm**

(LCA) showed more modest results. The Hyperbolic Spiral, despite its strengths in diversification and intensification, recorded an average recall of **68.56**, whereas the LCA lagged slightly behind with **69.86**, marking areas for improvement in these algorithms to better address the complexities of wildfire prediction in Saskatchewan.

Statistical Validation: The Friedman test validated the performance differences, yielding a test statistic of **76.04** and a p-value of **4.18e-09**, confirming significant differences in algorithm effectiveness. Figure 4.7 visualizes the rankings, showcasing the consistent superiority of spiral-based enhancements over the standard LCA.

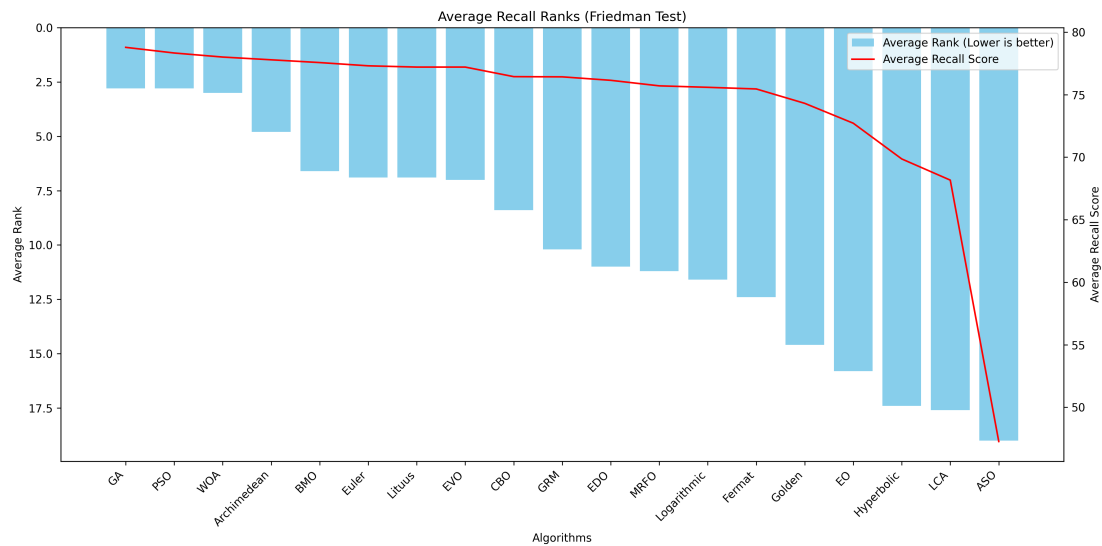


Figure 4.7: Visualization of average recall rankings among algorithms in Saskatchewan (Friedman Test results). Lower ranks indicate better performance, with spiral-enhanced algorithms generally outperforming the LCA.

Algorithm Performance in Yukon

In Yorkton, the **Genetic Algorithm (GA)** again topped the performance charts with an outstanding average recall of **81.80**, proving its consistent effectiveness across different geographical datasets. The **Particle Swarm Optimization (PSO)** was also notably effective, scoring an average recall of **80.90**, particularly excelling in the **Gradient Boosting** and **Logistic Regression** models.

The **Atom Search Optimization (ASO)**, much like in Manitoba, recorded the lowest performance in Yorkton with an average recall of **51.81**. This continued underperformance across regions suggests a need for revisiting the adaptation of ASO to regional wildfire data specifics. Contrarily, **PSO** and **EDO** showed remarkable adaptability and effectiveness, marked by high recall scores across the board, thus asserting their utility in high-stakes environments such as wildfire prediction.

This regional analysis underlines the critical importance of choosing the right algorithm based on both the nature of the data and specific regional characteristics to optimize performance outcomes in wildfire prediction tasks.

Table 4.14: Algorithm Rankings for Yorkton

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
GA	82.40	84.86	72.24	90.37	79.12	13	2.60	1
PSO	81.58	85.87	67.83	89.34	79.90	18	3.60	2
BMO	81.10	85.70	65.17	90.67	72.95	25	5.00	3
EDO	78.77	90.12	68.98	86.27	74.17	25	5.00	3
MRFO	79.79	79.80	69.28	87.78	74.61	26	5.20	5
CBO	78.76	80.91	67.58	88.93	80.53	27	5.40	6
EVO	77.51	83.00	64.47	90.30	73.76	35	7.00	7
WOA	75.57	76.50	67.35	90.44	73.57	38	7.60	8
EO	77.94	78.37	72.72	79.12	70.13	39	7.80	9
GRM	78.71	77.59	69.02	83.01	70.24	40	8.00	10
LCA	74.61	76.33	71.09	82.89	71.20	44	8.80	11
ASO	50.94	55.30	48.95	51.94	51.91	60	12.00	12

Spiral-Based Enhancements: The assessment of spiral-based algorithms in Yorkton presents intriguing findings on their efficacy in wildfire prediction, as detailed in Table [4.14](#).

Leading the evaluation, the **Euler Spiral** stood out significantly, achieving the top average recall of **81.07**. It showcased exceptional performance across multiple models, especially notable in **Logistic Regression** where it reached a recall of **91.85**, and in **Gradient Boosting** with **85.71**. This underscores the Euler Spiral's robust adaptability and superior capability in managing complex datasets effectively.

The **Archimedean Spiral** secured the second rank with a strong average recall of **78.93**. It performed particularly well in **Logistic Regression**, achieving the highest recall in this model at **93.11**. Its consistent performance across other models also highlights its reliable predictive power.

Following closely, the **Logarithmic Spiral** recorded a solid average recall of **78.42**, excelling in **Logistic Regression** with a recall of **87.08**. This spiral's systematic approach to data exploration and extraction of meaningful patterns contributes significantly to its high performance.

The **Hyperbolic Spiral** ranked fourth with an average recall of **76.39**, demonstrating strong capability in **Logistic Regression** with a recall of **84.86**. Its unique method of exploring the search space ensures effective feature selection, albeit slightly behind the top performers.

Sharing the fifth rank, both the **Fermat Spiral** and the **Liver Cancer Algorithm (LCA)** achieved an average recall of **75.62**. The Fermat Spiral performed well across models with particularly high scores in **Logistic Regression**, whereas

the LCA showed balanced performance, indicating their potential in specific contexts.

On the lower end, the **Golden Spiral** and **Lituus Spiral** encountered challenges, with average recalls of **69.12** and **59.02** respectively. These results suggest that these algorithms may require further refinement to enhance their predictive accuracy in the context of Yorkton’s complex wildfire dynamics.

Table 4.15: Spiral Algorithm Rankings for Yorkton

Algorithm	Random Forest	Gradient Boosting	K-Nearest Neighbor	Logistic Regression	Support Vector Machine	Rank Sum	Average Ranking	Ranking
Euler	85.30	85.71	70.98	91.85	75.49	10	2.00	1
Archimedean	78.29	78.77	67.76	93.11	76.72	14	2.80	2
Logarithmic	79.90	78.88	70.90	87.08	75.42	15	3.00	3
Hyperbolic	78.06	74.91	71.05	84.86	73.05	21	4.20	4
LCA	74.61	76.33	71.09	82.89	71.20	22	4.40	5
Fermat	74.46	75.37	73.12	83.74	69.42	23	4.60	6
Golden	70.87	68.26	65.39	78.78	62.32	35	7.00	7
Lituus	58.99	56.46	56.24	70.31	57.10	40	8.00	8

Statistical Validation: The Friedman test confirmed significant differences in algorithm performance, yielding a test statistic of **55.49** and a p-value of **1.07e-05**. Figure 4.8 visualizes the rankings, with spiral-based algorithms generally outperforming the standard **LCA**.

4.2 Provincial Result Summary

This study evaluated multiple metaheuristic algorithms for wildfire prediction across various Canadian provinces, highlighting their strengths and limitations in different regional contexts. By focusing on **Recall** as the primary metric, we prioritized minimizing false negatives to identify high-risk wildfire cases accurately. The results demonstrated that algorithms such as the **Energy Valley Optimizer (EVO)**,

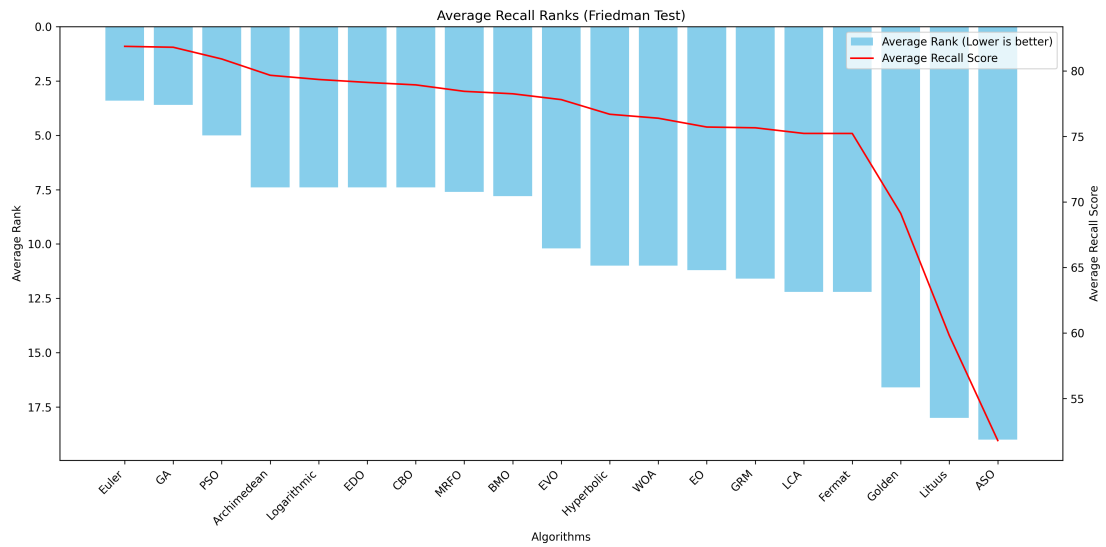


Figure 4.8: Visualization of average recall rankings among algorithms in Yukon (Friedman Test results). Lower ranks indicate better performance, with spiral-enhanced algorithms demonstrating significant improvements.

Genetic Algorithm (GA), and **Golden Ratio Method (GRM)** consistently outperformed others in most provinces, showcasing their robustness and adaptability to diverse environmental conditions. Conversely, standard algorithms like the **Atom Search Optimization (ASO)** and **Whale Optimization Algorithm (WOA)** consistently struggled to adapt to the complexities of wildfire prediction, resulting in lower recall scores across provinces. Similarly, **Logistic Regression** was the weakest predictive model overall, consistently delivering suboptimal recall values. On the other hand, **Random Forest** emerged as the most reliable model for Recall across top-ranking algorithms, particularly when combined with advanced feature selection techniques.

Wilcoxon Signed-Rank Test To evaluate the effectiveness of various spiral-based algorithms relative to the Liver Cancer Algorithm (LCA), we performed

Wilcoxon signed-rank tests. This non-parametric test was employed to compare matched-pairs of algorithm performance scores, aiming to determine if there are statistically significant differences between the LCA and each of the compared algorithms.

The heatmap displayed below summarizes the Wilcoxon statistic values and their corresponding p-values for each spiral algorithm comparison:

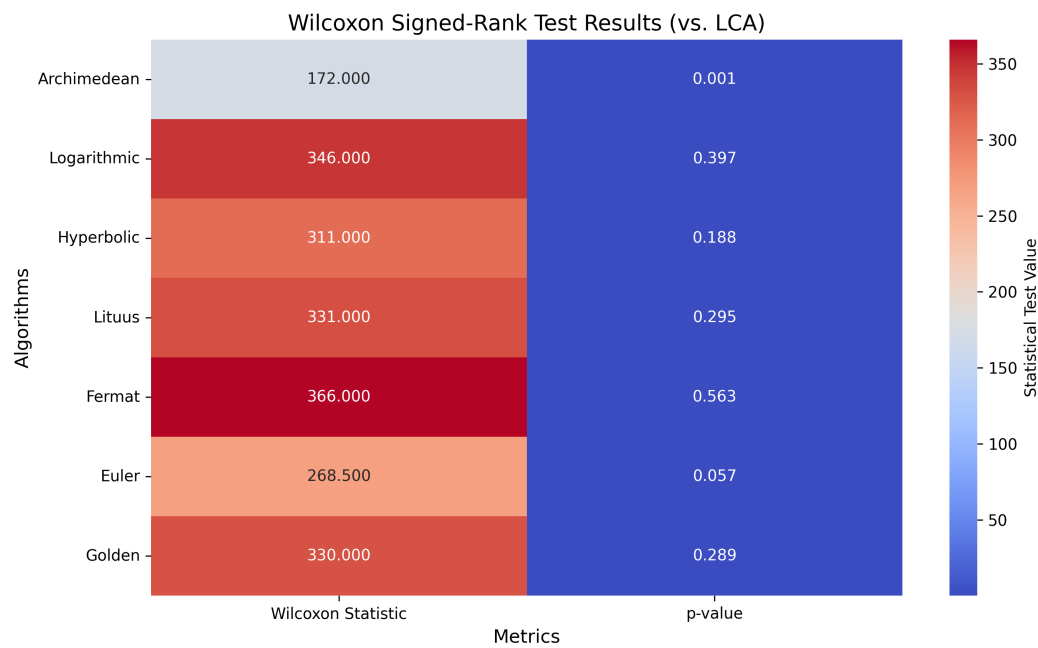


Figure 4.9: Wilcoxon Signed-Rank Test Results vs. LCA

- **Wilcoxon Statistic:** This value is indicative of the magnitude of differences between the matched pairs. A higher statistic suggests a greater disparity in performance ranks between LCA and the compared algorithm.
- **P-value:** This metric assesses the statistical significance of the observed differences. A p-value less than 0.05 typically indicates that the differences

are statistically significant, leading us to reject the null hypothesis of no difference.

Key Findings

- The **Archimedean Spiral** showed significant differences from the LCA, with a p-value of **0.001**, indicating a substantial improvement over the LCA.
- Conversely, algorithms like the **Fermat Spiral** and **Logarithmic Spiral** displayed higher p-values (**0.563** and **0.397** respectively), suggesting that the performance differences from the LCA are not statistically significant.
- The **Euler Spiral** exhibited a borderline p-value of **0.057**, pointing to a potential significant difference, but still requires further investigation or a larger sample size to confirm this finding.

4.2.1 Key Insights Across Provinces

Top-Performing Algorithms:

- The **Energy Valley Optimizer (EVO)**, **Genetic Algorithm (GA)**, and **Particle Swarm Optimization (PSO)** were among the most robust algorithms across multiple provinces. For example, in Saskatchewan, the **Genetic Algorithm** and **PSO** achieved average ranks of **2.8**, showcasing their effectiveness in wildfire prediction tasks.
- Spiral-enhanced methods such as the **Euler Spiral** and **Archimedean Spiral** were consistently ranked as superior by both Friedman and Wilcoxon

tests when compared to the standard **Liver Cancer Algorithm (LCA)**. These methods demonstrated particularly strong performance in regions like Alberta and Yukon, achieving high recall scores in models like **Random Forest** and **Gradient Boosting**.

Algorithm Variability Across Provinces:

- Regional factors significantly influenced algorithm performance. While the **Whale Optimization Algorithm (WOA)** performed moderately well in Alberta, it ranked lower in provinces like Ontario and Quebec, indicating its sensitivity to regional wildfire patterns.
- Similarly, the **Golden Ratio Method (GRM)** excelled in provinces like Yukon but struggled in Manitoba and Northwest Territories, highlighting the need for adaptable optimization strategies.
- There might be a link between location and algorithm performance as all locations share the same variables. The differences in the best algorithm performance in each location suggest that regional characteristics and data distributions can significantly impact outcomes, pointing towards the need for location-specific tuning of algorithms.

Importance of Feature Selection: Feature selection played a critical role in enhancing predictive accuracy across all regions. Table 4.16 presents the ten most frequently selected features, reflecting their importance in wildfire prediction tasks. Notably, **Cluster Density Risk Score**, **Fire Radiative Power**, and **Temperature at 2 Meters Celsius** emerged as universally significant, emphasizing their relevance across diverse wildfire conditions.

Table 4.16: Top 10 Most Selected Features by All Algorithms

Rank	Feature Name	Selection Frequency
1	Cluster Density Risk Score (Feature 11)	95%
2	Fire Radiative Power (Feature 0)	90%
3	Temperature at 2 Meters Celsius (Feature 29)	85%
4	Total Cloud Cover (Feature 6)	82%
5	Temperature-Wind Interaction (Feature 43)	80%
6	Week (Temporal Feature) (Feature 17)	78%
7	High Vegetation Fire Risk Score (Feature 21)	75%
8	Combined High Vegetation (Feature 20)	72%
9	Leaf Area Index for High Vegetation (Feature 9)	70%
10	Surface Pressure (Feature 8)	68%

Chapter 5

Conclusion

This study has significantly contributed to wildfire prediction by integrating advanced metaheuristic feature selection techniques, refining the Liver Cancer Algorithm (LCA) with spiral updates, and identifying critical environmental predictors of wildfire risk across Canadian provinces. Through innovative methodology and rigorous analysis, we have addressed essential challenges of wildfire modelling and provided insights that can inform research and practical applications.

5.1 Key Advancements

One of the primary advancements of this study is the use of metaheuristic algorithms for feature selection. These algorithms reduce the dimensionality of complex datasets, allowing predictive models to focus on the most influential variables. By doing so, we have enhanced the accuracy of wildfire predictions across diverse provincial contexts. The importance of feature selection in wildfire modelling has

been well-documented in the existing literature. For instance, Dong et al. (2022) highlighted the critical role of spatiotemporal factors, such as temperature and vegetation indices, in determining wildfire occurrence Dong et al. [2022]. Similarly, Wang et al. (2021) emphasized the interplay of local meteorology and land-surface characteristics in shaping wildfire dynamics Wang et al. [2021]. Our findings align closely with these studies, demonstrating the effectiveness of feature selection in capturing the complex interdependencies that drive wildfire behaviour.

The refinement of the LCA through spiral updates represents another critical contribution. Traditional optimization algorithms often struggle to balance exploration and exploitation within the search space, leading to suboptimal solutions. By incorporating spiral dynamics, particularly the Euler Spiral, we have introduced a more efficient mechanism for navigating the solution space. This innovation has improved the LCA’s convergence rate and enhanced its stability and precision in feature selection. While regional differences in wildfire predictors were expected due to environmental variability, a notable finding was the consistent selection of critical features across provinces, observed in spiral-enhanced and standard algorithms. This underscores the robustness of the identified features and their universal relevance in wildfire modelling.

5.2 Environmental Variability

A key observation of this study was that despite regional differences in wildfire characteristics, the spiral-enhanced and standard algorithms consistently identified similar influential features across provinces. This finding suggests that certain environmental factors, such as temperature, vegetation density, and fire radiative power, play a universal role in wildfire dynamics. For instance, the consistent

selection of Cluster Density Risk Score across provinces indicates that spatial clustering patterns are critical to understanding wildfire behaviour, regardless of regional variability. These results highlight the adaptability of the algorithms and reinforce the robustness of the identified predictors in diverse environmental contexts.

5.3 Computational Constraints

While this study demonstrated the effectiveness of metaheuristic algorithms, computational efficiency remains a challenge. The large datasets used in this research required substantial computational resources, particularly for implementing cross-validation and running multiple algorithm iterations. These constraints limited the scalability of the approach, especially for real-time applications where rapid predictions are essential. Addressing these challenges is vital for operationalizing the methods proposed in this study. Optimizing the computational framework or employing distributed processing systems could enhance the scalability and efficiency of these algorithms.

5.4 Future Work

The findings of this study underscore the importance of not only pinpointing the occurrence of wildfires but also recognizing conditions that denote high risk yet do not culminate in fire events. This insight directs us toward multiple fruitful avenues for future research:

1. **Inclusion of Non-Event High-Risk Days:** Future datasets could incorpo-

rate days characterized as high-risk based on environmental and topographical data but where no wildfires occurred. Analyzing these instances will aid in distinguishing false positives and refining the conditions under which wildfires are predicted, thereby enhancing the model's discriminatory power between actual and potential fire scenarios.

2. **Development of Ensemble and Hybrid Methods:** Extending the capabilities of ensemble methods that amalgamate the strengths of multiple metaheuristic algorithms, future studies could also explore hybrid models that integrate these techniques with data from non-event high-risk days. Approaches such as combining spiral dynamics with other optimization strategies like genetic algorithms or swarm intelligence could further the efficacy and reliability of wildfire prediction models.
3. **Region-Specific Algorithm Tuning:** The observed variability in algorithm performance across different locations indicates a significant impact of regional characteristics on wildfire predictions. Future research could concentrate on customizing algorithms to specific locales by adapting algorithm parameters or integrating regional data traits, including those from high-risk non-event days. This tailored approach aims to refine the predictive accuracy and robustness of models designed for distinct geographic areas.
4. **Investigating Preventive Measures:** By incorporating data from days deemed high-risk but without wildfire occurrences, future research can also evaluate the impact of existing wildfire prevention and mitigation strategies. Insights into what effectively prevents wildfires on these high-risk days could inform more precise preventive measures and resource allocation strategies.

Pursuing these research avenues could significantly advance the precision, relia-

bility, and contextual suitability of wildfire prediction models, thereby contributing to enhanced strategic planning and risk management in wildfire-prone regions.

5.5 Closing Remarks

This study represents a meaningful step forward in the field of wildfire prediction. By integrating advanced feature selection techniques, refining optimization algorithms, and identifying critical environmental predictors, we have provided a comprehensive framework for understanding and managing wildfire risks. These contributions advance scientific knowledge and have practical implications for mitigating the devastating impacts of wildfires on ecosystems, economies, and communities. As climate change exacerbates wildfire activity, the need for accurate and reliable prediction models has never been more urgent. This study serves as a foundation for future research and innovation, paving the way for more effective wildfire management strategies in the years to come.

Bibliography

- Stephen J. Pyne, Patricia L. Andrews, and Richard D. Laven. *Introduction to wildland fire*. Wiley, 1996. ISBN 0471549134. URL <https://ezproxy.tru.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat03106a&AN=tru.a98336&site=eds-live&scope=site>.
- Cristina Santín and Stefan H. Doerr. Fire effects on soils: the human dimension. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371, 6 2016. ISSN 14712970. doi: 10.1098/RSTB.2015.0171. URL [/pmc/articles/PMC4874409//pmc/articles/PMC4874409/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC4874409/](https://pmc/articles/PMC4874409//pmc/articles/PMC4874409/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC4874409/).
- Ioannis Prapas, Spyros Kondylatos, Ioannis Papoutsis, Gustau Camps-Valls, Michele Ronco, Miguel Ángel Fernández-Torres, Maria Piles Guillem, and Nuno Carvalhais. Deep learning methods for daily wildfire danger forecasting, 2021.
- Alexandra Tyukavina, Peter Potapov, Matthew C. Hansen, Amy H. Pickens, Stephen V. Stehman, Svetlana Turubanova, Diana Parker, Viviana Zalles, André Lima, Indrani Kommareddy, Xiao Peng Song, Lei Wang, and Nancy Harris. Global trends of forest loss due to fire from 2001 to 2019. *Frontiers in Remote Sensing*, 3:825190, 3 2022. ISSN 26736187. doi: 10.3389/FRSEN.2022.825190/BIBTEX.

Peter Potapov, Matthew C. Hansen, Lars Laestadius, Svetlana Turubanova, Alexey Yaroshenko, Christoph Thies, Wynet Smith, Ilona Zhuravleva, Anna Komarova, Susan Minnemeyer, and Elena Esipova. The last frontiers of wilderness: Tracking loss of intact forest landscapes from 2000 to 2013. *Science advances*, 3, 1 2017. ISSN 2375-2548. doi: 10.1126/SCIADV.1600821. URL <https://pubmed.ncbi.nlm.nih.gov/28097216/>.

Mikaela Weisse, Elizabeth Goldman, and Sarah Carter. The latest analysis on global forests and tree cover loss global forest review, 2022. URL <https://research.wri.org/gfr/latest-analysis-deforestation-trends>.

Government Of Canada. Canada’s record-breaking wildfires in 2023: A fiery wake-up call, 2024. URL <https://natural-resources.canada.ca/simply-science/canadas-record-breaking-wildfires-2023-fiery-wake-call/25303>.

CIFFCI. Cifc — wildfire graphs, 2024. URL <https://ciffc.net/statistics>.

Luke P. Naeher, Michael Brauer, Michael Lipsett, Judith T. Zelikoff, Christopher D. Simpson, Jane Q. Koenig, and Kirk R. Smith. Woodsmoke health effects: a review. *Inhalation toxicology*, 19:67–106, 1 2007. ISSN 1091-7691. doi: 10.1080/08958370600985875. URL <https://pubmed.ncbi.nlm.nih.gov/17127644/>.

Colleen E. Reid, Michael Brauer, Fay H. Johnston, Michael Jerrett, John R. Balmes, and Catherine T. Elliott. Critical review of health impacts of wildfire smoke exposure. *Environmental Health Perspectives*, 124(9):1334–1343, 2016. doi: 10.1289/ehp.1409277. URL <https://ehp.niehs.nih.gov/doi/abs/10.1289/ehp.1409277>.

Raj P. Fadadu, Barbara Grimes, Nicholas P. Jewell, Jason Vargo, Albert T. Young, Katrina Abuabara, John R. Balmes, and Maria L. Wei. Association of wildfire air

- pollution and health care use for atopic dermatitis and itch. *JAMA Dermatology*, 157:658–666, 6 2021. ISSN 21686084. doi: 10.1001/jamadermatol.2021.0179.
- Terry L. Noah, Cameron P. Worden, Meghan E. Rebuli, and Ilona Jaspers. The effects of wildfire smoke on asthma and allergy. *Current allergy and asthma reports*, 23:375–387, 7 2023. ISSN 1534-6315. doi: 10.1007/S11882-023-01090-1. URL <https://pubmed.ncbi.nlm.nih.gov/37171670/>.
- Kathyana P. Santiago Mangual, Sarah Ferree, Jenny E. Murase, and Arianne Shadi Kourosch. The burden of air pollution on skin health: a brief report and call to action. *Dermatology and therapy*, 14:251–259, 1 2024. ISSN 2193-8210. doi: 10.1007/S13555-023-01080-1. URL <https://pubmed.ncbi.nlm.nih.gov/38103119/>.
- Fabienne Reisen, Sandra M. Duran, Mike Flannigan, Catherine Elliott, and Karen Rideout. Wildfire smoke and public health risk, 2015. ISSN 10498001.
- Daniel A. Jaffe, Susan M. O’Neill, Narasimhan K. Larkin, Amara L. Holder, David L. Peterson, Jessica E. Halofsky, and Ana G. Rappold. Wildfire and prescribed burning impacts on air quality in the united states. *Journal of the Air & Waste Management Association (1995)*, 70:583, 6 2020. ISSN 21622906. doi: 10.1080/10962247.2020.1749731. URL [/pmc/articles/PMC7932990/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7932990/)[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7932990/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7932990/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC7932990/).
- Laura A. Burkle, Jonathan A. Myers, R. Travis Belote, and D. P.C. Peters. Wildfire disturbance and productivity as drivers of plant species diversity across spatial scales. *Ecosphere*, 6, 10 2015. ISSN 21508925. doi: 10.1890/ES15-00438.1.
- Laura J. Heil and Laura A. Burkle. Recent post-wildfire salvage logging benefits

- local and landscape floral and bee communities. *Forest Ecology and Management*, 424:267–275, 9 2018. ISSN 03781127. doi: 10.1016/j.foreco.2018.05.009.
- Daisy Gates, Breeanne Jackson, and Sean D. Schoville. Impacts of fire on butterfly genetic diversity and connectivity. *Journal of Heredity*, 112:367–376, 6 2021. ISSN 14657333. doi: 10.1093/jhered/esab027.
- Jessica E. Halofsky, David L. Peterson, and Brian J. Harvey. Changing wildfire, changing forests: the effects of climate change on fire regimes and vegetation in the pacific northwest, usa. *Fire Ecology*, 16:1–26, 12 2020. ISSN 19339747. doi: 10.1186/S42408-019-0062-8/FIGURES/4. URL <https://fireecology.springeropen.com/articles/10.1186/s42408-019-0062-8>.
- Fatih Sari. Identifying anthropogenic and natural causes of wildfires by maximum entropy method-based ignition susceptibility distribution models. *Journal of Forestry Research*, 34:355–371, 4 2023. ISSN 19930607. doi: 10.1007/S11676-022-01502-4/TABLES/8. URL <https://link.springer.com/article/10.1007/s11676-022-01502-4>.
- Cordy Tymstra, Brian J. Stocks, Xinli Cai, and Mike D. Flannigan. Wildfire management in canada: Review, challenges and opportunities. *Progress in Disaster Science*, 5:100045, 1 2020. ISSN 2590-0617. doi: 10.1016/J.PDISAS.2019.100045.
- Sean C.P. Coogan, Lori D. Daniels, Den Boychuk, Philip J. Burton, Mike D. Flannigan, Sylvie Gauthier, Victor Kafka, Jane S. Park, and B. Mike Wotton. Fifty years of wildland fire science in canada. *Canadian Journal of Forest Research*, 51:283–302, 2021. ISSN 12086037. doi: 10.1139/CJFR-2020-0314/ASSET/IMAGES/LARGE/CJFR-2020-0314F5.JPEG. URL <https://cdnsiencepub.com/doi/10.1139/cjfr-2020-0314>.

- M. D. Flannigan, K. A. Logan, B. D. Amiro, W. R. Skinner, and B. J. Stocks. Future area burned in canada. *Climatic Change*, 72:1–16, 9 2005. ISSN 01650009. doi: 10.1007/S10584-005-5935-Y/METRICS. URL <https://link.springer.com/article/10.1007/s10584-005-5935-y>.
- Christelle Hély, Mike Flannigan, Yves Bergeron, and Douglas McRae. Role of vegetation and weather on fire behavior in the canadian mixedwood boreal forest using two fire behavior prediction systems. *Can. J. For. Res.*, 31:430–441, 2001. ISSN 0045-5067. doi: 10.1139/cjfr-31-3-430.
- Tzeidle N Wasserman and Stephanie E Mueller. Review open access climate influences on future fire severity: a synthesis of climate-fire interactions and impacts on fire regimes, high-severity fire, and forests in the western united states. *Fire Ecology*, 2023. doi: 10.1186/s42408-023-00200-8. URL <http://creativecommons.org/licenses/by/4.0/.FireEcology>.
- Mohammad Reza Alizadeh, John T. Abatzoglou, Charles H. Luce, Jan F. Adamowski, Arvin Farid, and Mojtaba Sadegh. Warming enabled upslope advance in western us forest fires. *Proceedings of the National Academy of Sciences of the United States of America*, 118, 6 2021. ISSN 10916490. doi: 10.1073/PNAS.2009717118/FORMAT/EPUB.
- Kira M. Hoffman, Amy Cardinal Christianson, Sarah Dickson-Hoyle, Kelsey Copes-Gerbitz, William Nikolakis, David A. Diabo, Robin McLeod, Herman J. Michell, Abdullah Al Mamun, Alex Zahara, Nicholas Mauro, Joe Gilchrist, Russell Myers Ross, and Lori D. Daniels. The right to burn: barriers and opportunities for indigenous-led fire stewardship in canada. *Facets*, 7:464–481, 2022. ISSN 23711671. doi: 10.1139/FACETS-2021-0062/ASSET/IMAGES/MEDIUM/FACETS-2021-0062F3.GIF. URL <https://www.facetsjournal.com/doi/10.1139/facets-2021-0062>.

- Mary R. Huffman. The many elements of traditional fire knowledge: Synthesis, classification, and aids to cross-cultural problem solving in fire-dependent systems around the world. *Ecology and Society*, 18(4), 2013. ISSN 17083087. URL <http://www.jstor.org/stable/26269435>.
- Amy Christianson. Social science research on indigenous wildfire management in the 21st century and future research needs. *International Journal of Wildland Fire*, 24:190–200, 2015. ISSN 10498001. doi: 10.1071/WF13048.
- Kenneth Johnstone. Timber and trauma. 75 years with the federal forestry service 1899-1974. *Forestry Canada*, 1991.
- Richard A Rajala. Feds, forests, and fire a century of canadian forestry innovation, 2005. URL www.sciencetech.technomuses.ca.
- B J Stocks², M E Alexander⁴, and C E Van. The canadian forest fire danger rating system: An overview'. *Forestry Canada*, 1989.
- P M Paul. Field practices in forest fire danger rating, 11 1969.
- RS McALPINE, BJ Stocks, Ce Vanwagner, BD LAWSON, Me Au, and T Lynham. Forest fire behavior research in canada. *Forest Fire Research*, 1990.
- Ce Van Wagner. Development and structure of the canadian forest fire weather index system. *Government Of Canada- Canadian Forestry Service*, 1987.
- William J De Groot. Interpreting the canadian forest fire weather index (fwi) systemi, 1987.
- Patricia L. Andrews. *The Rothermel surface fire spread model and associated developments: A comprehensive explanation*. U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2018. doi: 10.2737/rmrs-gtr-371. URL <http://dx.doi.org/10.2737/RMRS-GTR-371>.

- C. Tymstra, R.W. Bryce, B.M. Wotton, S.W. Taylor, and O.B. Armitage. Development and structure of prometheus: the canadian wildland fire growth simulation model, available at: <https://cfs.nrcan.gc.ca/publications?id=31775>. *Canadian Forest Service*, NOR-X-417, 2010. ISSN 0831-8247.
- J Barber, C Bose, A Bourlioux, J Braun, E Brunelle, T Garcia, ¶ T Hillen, and B Ong. Burning issues with prometheus-the canadian wildland fire growth simulation model, 2009.
- M D Flannigan and J B Harrington. A study of the relation of meteorological variables to monthly provincial area burned by wildfire in canada (1953–80). *Journal of Applied Meteorology and Climatology*, 27:441–452, 1988. doi: 10.1175/1520-0450(1988)027<0441:ASOTRO>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/apme/27/4/1520-0450_1988_027_0441_asotro_2_0_co_2.xml.
- B. Mike Wotton. Interpreting and using outputs from the canadian forest fire danger rating system in research applications. *Environmental and Ecological Statistics*, 16:107–131, 3 2009. ISSN 13528505. doi: 10.1007/S10651-007-0084-2/METRICS. URL <https://link.springer.com/article/10.1007/s10651-007-0084-2>.
- W. Matt Jolly, Mark A. Cochrane, Patrick H. Freeborn, Zachary A. Holden, Timothy J. Brown, Grant J. Williamson, and David M.J.S. Bowman. Climate-induced variations in global wildfire danger from 1979 to 2013. *Nature Communications*, 6, 7 2015. ISSN 20411723. doi: 10.1038/ncomms8537.
- M S BALSHI, A D MCGUIRE, P DUFFY, M FLANNIGAN, D W KICKLIGHTER, and J MELILLO. Vulnerability of carbon storage in north american boreal forests to wildfires during the 21st century. *Global Change Biology*, 15: 1491–1510, 6 2009. ISSN 1354-1013. doi: <https://doi.org/10.1111/j.1365-2486>.

2009.01877.x. URL <https://doi.org/10.1111/j.1365-2486.2009.01877.x>.
<https://doi.org/10.1111/j.1365-2486.2009.01877.x>.

David M Romps, Jacob T Seeley, David Vollaro, and John Molinari. Projected increase in lightning strikes in the united states due to global warming. *Science*, 346:851–854, 11 2014. doi: 10.1126/science.1259100. URL <https://doi.org/10.1126/science.1259100>. doi: 10.1126/science.1259100.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Vasilev Ivan, Slater Daniel, Spacagna Gianmario, Roelants Peter, and Zocca Valentino. *Python Deep Learning : Exploring Deep Learning Techniques and Neural Network Architectures with PyTorch, Keras, and TensorFlow.*, volume Second edition. Packt Publishing, 2019. ISBN 9781789348460. URL <https://ezproxy.tru.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2002295&site=eds-live&scope=site>.

Tanya. Kolosova and Samuel. Berestizhevsky. *Supervised Machine Learning Optimization Framework and Applications with SAS and R*. CRC Press LLC, 1 edition, 2020. ISBN 9780367277321.

Blaine Bateman, Ashish Jha, Benjamin Johnston, Ishita Mathur, and an O'Reilly Media Company. Safari. *The Supervised Learning Workshop - Second Edition*. Packt Publishing, Limited, 2020. ISBN 9781800209046.

Hinton Geoffrey and Sejnowski Terrence J. *Unsupervised Learning : Foundations of Neural Computation*. Computational Neuroscience. A Bradford Book, 1999. ISBN 9780262581684. URL <https://ezproxy.tru.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=48890&site=eds-live&scope=site>.

Sutton Richard S. and Barto Andrew G. *Reinforcement Learning : An Introduction*. Adaptive Computation and Machine Learning. Bradford Books, 1998. ISBN 9780262193986. URL <https://ezproxy.tru.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1094&site=eds-live&scope=site>.

R. H. Nolan, M. M. Boer, V. Resco de Dios, G. Caccamo, and R. A. Bradstock. Large-scale, dynamic transformations in fuel moisture drive wildfire activity across southeastern australia. *Geophysical Research Letters*, 43(9):4229–4238, 2016. doi: <https://doi.org/10.1002/2016GL068614>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016GL068614>.

Christelle Heacute;ly, Mike Flannigan, Yves Bergeron, and Douglas McRae. Role of vegetation and weather on fire behavior in the canadian mixedwood boreal forest using tow fire behavior prediction systems. *Canadian Journal of Forest Research*, 31(3):430, 2001. ISSN 00455067. URL <https://ezproxy.tru.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=8745393&site=eds-live&scope=site>.

Chuvieco Emilio. *Wildland Fire Danger Estimation And Mapping: The Role Of Remote Sensing Data*. Number v. 4 in Series in Remote Sensing. World Scientific, 2003. ISBN 9789812385697. URL <https://ezproxy.tru.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=514685&site=eds-live&scope=site>.

Liujun Zhu, Geoffrey I. Webb, Marta Yebra, Gianluca Scortechini, Lynn Miller, and Francois Petitjean. Live fuel moisture content estimation from modis: A deep learning approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 179:81–91, 9 2021. ISSN 0924-2716. doi: 10.1016/J.ISPRSJPRS.2021.07.010.

- Jose Rivera, Daniel San Martin, Michael Gollner, Claudio E. Torres, and Carlos Fernandez-Pello. A machine learning approach to predict the critical heat flux for ignition of solid fuels. *Fire Safety Journal*, 141, 12 2023. ISSN 03797112. doi: 10.1016/j.firesaf.2023.103968.
- Àngel Cunill Camprubí, Pablo González-Moreno, and Víctor Resco de Dios. Live fuel moisture content mapping in the mediterranean basin using random forests and combining modis spectral and thermal data. *Remote Sensing*, 14(13), 2022. ISSN 2072-4292. doi: 10.3390/rs14133162. URL <https://www.mdpi.com/2072-4292/14/13/3162>.
- José M. Costa-Saura, Ángel Balaguer-Beser, Luis A. Ruiz, Josep E. Pardo-Pascual, and José L. Soriano-Sancho. Empirical models for spatio-temporal live fuel moisture content estimation in mixed mediterranean vegetation areas using sentinel-2 indices and meteorological data. *Remote Sensing*, 13:1–26, 9 2021. ISSN 2072-4292. doi: 10.3390/RS13183726. URL <https://riunet.upv.es/handle/10251/184451>.
- Katherine Hayes, Chad M. Hoffman, Rodman Linn, Justin Ziegler, and Brian Buma. Fuel constraints, not fire weather conditions, limit fire behavior in reburned boreal forests. *Agricultural and Forest Meteorology*, 358:110216, 11 2024. ISSN 0168-1923. doi: 10.1016/J.AGRFORMET.2024.110216.
- Abolfazl Abdollahi and Marta Yebra. Forest fuel type classification: Review of remote sensing techniques, constraints and future trends, 9 2023. ISSN 10958630.
- Melissa A. Boyd, Xanthe J. Walker, Jennifer Barnes, Gerardo Celis, Scott J. Goetz, Jill F. Johnstone, Nicholas T. Link, April M. Melvin, Lisa Saperstein, Edward A.G. Schuur, and Michelle C. Mack. Decadal impacts of wildfire fuel reduction treatments on ecosystem characteristics and fire behavior in alaskan

boreal forests. *Forest Ecology and Management*, 546:121347, 10 2023. ISSN 0378-1127. doi: 10.1016/J.FORECO.2023.121347.

Usda Forest Service Rocky Mountain Research Station. Estimating wildfire behavior and effects. *Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Intermountain Forest and Range Experiment Station. 40 p.*, 1976.

Usda Forest Service Rocky Mountain Research Station. A mathematical model for predicting fire spread in wildland fuels. *Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Intermountain Forest and Range Experiment Station. 40 p.*, 115, 1972. URL <https://research.fs.usda.gov/treesearch/32533>.

Viktor Myroniuk, Sergiy Zibtsev, Vadym Bogomolov, Johann Georg Goldammer, Oleksandr Soshenskyi, Viacheslav Levchenko, and Maksym Matsala. Combining landsat time series and gedi data for improved characterization of fuel types and canopy metrics in wildfire simulation. *Journal of Environmental Management*, 345:118736, 11 2023. ISSN 0301-4797. doi: 10.1016/J.JENVMAN.2023.118736.

Joel McCorkel, Matthew Montanaro, Boryana Efremova, Aaron Pearlman, Brian Wenny, Allen Lunsford, Amy Simon, Jason Hair, and Dennis Reuter. Landsat 9 thermal infrared sensor 2 characterization plan overview. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 8845–8848, 2018. doi: 10.1109/IGARSS.2018.8518798.

Jakob Sigurdsson, Sveinn E. Armannsson, Magnus O. Ulfarsson, and Johannes R. Sveinsson. Fusing sentinel-2 and landsat 8 satellite images using a model-based method. *Remote Sensing 2022, Vol. 14, Page 3224*, 14:3224, 7 2022. ISSN 2072-4292. doi: 10.3390/RS14133224. URL <https://www.mdpi.com/2072-4292/14/13/3224/htm><https://www.mdpi.com/2072-4292/14/13/3224>.

Martin Claverie, Junchang Ju, Jeffrey G. Masek, Jennifer L. Dungan, Eric F. Vermote, Jean Claude Roger, Sergii V. Skakun, and Christopher Justice. The harmonized landsat and sentinel-2 surface reflectance data set. *Remote Sensing of Environment*, 219:145–161, 12 2018. ISSN 00344257. doi: 10.1016/j.rse.2018.09.002.

Harmonized landsat and sentinel-2 (hls) — earthdata, 2023a. URL <https://www.earthdata.nasa.gov/esds/harmonized-landsat-sentinel-2>.

Sanjiwana Arjasakusuma, Sandiaga Swahyu Kusuma, Yenni Vetrita, Indah Prasasti, and Rahmat Arief. Monthly burned-area mapping using multi-sensor integration of sentinel-1 and sentinel-2 and machine learning: Case study of 2019’s fire events in south sumatra province, indonesia. *Remote Sensing Applications: Society and Environment*, 27:100790, 8 2022. ISSN 2352-9385. doi: 10.1016/J.RSASE.2022.100790.

Daniel Martin Nelson, Yuhong He, and G. W.K. Moore. Trends and applications in wildfire burned area mapping: Remote sensing data, cloud geoprocessing platforms, and emerging algorithms, 7 2024. ISSN 19254296.

Ehsan Khankeshizadeh, Sahand Tahermanesh, Amin Mohsenifar, Armin Moghimi, and Ali Mohammadzadeh. Fba-dpattresu-net: Forest burned area detection using a novel end-to-end dual-path attention residual-based u-net from post-fire sentinel-1 and sentinel-2 images. *Ecological Indicators*, 167, 10 2024. ISSN 1470160X. doi: 10.1016/j.ecolind.2024.112589.

Hasan Tonbul, Ismail Colkesen, and Taskin Kavzoglu. Pixel- and object-based ensemble learning for forest burn severity using usgs firemon and mediterranean condition dnbrs in aegean ecosystem (turkey). *Advances in Space Research*, 69: 3609–3632, 5 2022. ISSN 18791948. doi: 10.1016/j.asr.2022.02.051.

- Xikun Hu, Puzhao Zhang, and Yifang Ban. Large-scale burn severity mapping in multispectral imagery using deep semantic segmentation models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:228–240, 2 2023. ISSN 0924-2716. doi: 10.1016/J.ISPRSJPRS.2022.12.026.
- Zachary Langford, Jitendra Kumar, and Forrest Hoffman. Wildfire mapping in interior alaska using deep neural networks on imbalanced datasets. *IEEE International Conference on Data Mining Workshops, ICDMW*, 2018-November: 770–778, 7 2018. ISSN 23759259. doi: 10.1109/ICDMW.2018.00116.
- Seyd Teymoor Seydi, Mahdi Hasanlou, and Jocelyn Chanussot. Burnt-net: Wildfire burned area mapping with single post-fire sentinel-2 data and deep learning morphological neural network. *Ecological Indicators*, 140, 7 2022. ISSN 1470160X. doi: 10.1016/j.ecolind.2022.108999.
- Sunil Thapa, Vishwas Sudhir Chitale, Sudip Pradhan, Bikram Shakya, Sundar Sharma, Smriety Regmi, Sameer Bajracharya, Shankar Adhikari, and Gauri Shankar Dangol. *Forest Fire Detection and Monitoring*, pages 147–167. Springer International Publishing, Cham, 2021. ISBN 978-3-030-73569-2. doi: 10.1007/978-3-030-73569-2_8. URL https://doi.org/10.1007/978-3-030-73569-2_8.
- Panagiotis Barmpoutis, Periklis Papaioannou, Kosmas Dimitropoulos, and Nikos Grammalidis. A review on early forest fire detection systems using optical remote sensing. *Sensors*, 20(22), 2020. ISSN 1424-8220. doi: 10.3390/s20226442. URL <https://www.mdpi.com/1424-8220/20/22/6442>.
- Nasa tracks wildfires from above to aid firefighters below - nasa, 2023b. URL <https://www.nasa.gov/missions/aqua/nasa-tracks-wildfires-from-above-to-aid-firefighters-below/>.

- Neil Burrows, Bruce Ward, and A. Robinson. Jarrah forest fire history from stem analysis and anthropological evidence. *Australian Forestry*, 58:7–16, 01 1995. doi: 10.1080/00049158.1995.10674636.
- A. A. Cunningham and D. L. Martell. A stochastic model for the occurrence of man-caused forest fires. *Canadian Journal of Forest Research*, 3(2):282–287, 1973. doi: 10.1139/x73-038. URL <https://doi.org/10.1139/x73-038>.
- S. W. Taylor, Douglas G. Woolford, C. B. Dean, and David L. Martell. Wildfire Prediction to Inform Fire Management: Statistical Science Challenges. *Statistical Science*, 28(4):586 – 615, 2013. doi: 10.1214/13-STS451. URL <https://doi.org/10.1214/13-STS451>.
- Cristina Vega-Garcia. Applying neural network technology to human-caused wildfire occurrence prediction. *AI Applications*, 10:9–18, 01 1996.
- Amparo Alonso-Betanzos, Oscar Fontenla-Romero, Bertha Guijarro-Berdiñas, Elena Hernández-Pereira, María Inmaculada Paz Andrade, Eulogio Jiménez, Jose Luis Legido Soto, and Tarsy Carballas. An intelligent system for forest fire risk prediction and fire fighting management in galicia. *Expert Systems with Applications*, 25:545–554, 11 2003. ISSN 0957-4174. doi: 10.1016/S0957-4174(03)00095-2.
- Christos Vasilakos, Kostas Kalabokidis, John Hatzopoulos, George Kallos, and Yiannis Matsinos. Integrating new methods and tools in fire danger rating. *International Journal of Wildland Fire*, 16:306–316, 01 2007. doi: 10.1071/WF05091.
- Ritaban Dutta, Jagannath Aryal, Aruneema Das, and Jamie B. Kirkpatrick. Deep cognitive imaging systems enable estimation of continental-scale fire incidence from climate data. *Scientific Reports 2013 3:1*, 3:1–4, 11 2013. ISSN 2045-2322. doi: 10.1038/srep03188. URL <https://www.nature.com/articles/srep03188>.

- George E. Sakr, Imad H. Elhajj, George Mitri, and Uchechukwu C. Wejinya. Artificial intelligence for forest fire prediction. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pages 1311–1316, 2010. doi: 10.1109/AIM.2010.5695809.
- George E. Sakr, Imad H. Elhajj, and George Mitri. Efficient forest fire occurrence prediction for developing countries using two weather parameters. *Engineering Applications of Artificial Intelligence*, 24:888–894, 8 2011. ISSN 0952-1976. doi: 10.1016/J.ENGAPPAI.2011.02.017.
- Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. In *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, volume 2, pages 1214–1217, Sep. 2005. doi: 10.1109/WCNM.2005.1544272.
- Guoli Zhang, Ming Wang, and Kai Liu. Deep neural networks for global wildfire susceptibility modelling. *Ecological Indicators*, 127, 8 2021. ISSN 1470160X. doi: 10.1016/j.ecolind.2021.107735.
- Daniela Stojanova, Andrej Kobler, Peter Ogrinc, Bernard Ženko, and Sašo Džeroski. Estimating the risk of fire outbreaks in the natural environment. *Data Mining and Knowledge Discovery*, 24:411–442, 3 2012. ISSN 13845810. doi: 10.1007/S10618-011-0213-2/METRICS. URL <https://link.springer.com/article/10.1007/s10618-011-0213-2>.
- Sandra Oliveira, Friderike Oehler, Jesús San-Miguel-Ayanz, Andrea Camia, and José M.C. Pereira. Modeling spatial patterns of fire occurrence in mediterranean europe using multiple regression and random forest. *Forest Ecology and Management*, 275:117–129, 7 2012. ISSN 0378-1127. doi: 10.1016/J.FORECO.2012.03.003.

- Mohammad Tavakol Sadrabadi and Mauro Sebastián Innocente. Vegetation cover type classification using cartographic data for prediction of wildfire behaviour. *Fire* 2023, Vol. 6, Page 76, 6:76, 2 2023. ISSN 2571-6255. doi: 10.3390/FIRE6020076. URL <https://www.mdpi.com/2571-6255/6/2/76/html><https://www.mdpi.com/2571-6255/6/2/76>.
- Víctor Fernández-García, David Beltrán-Marcos, José Manuel Fernández-Guisuraga, Elena Marcos, and Leonor Calvo. Predicting potential wildfire severity across southern europe with global data sources. *Science of The Total Environment*, 829:154729, 7 2022. ISSN 0048-9697. doi: 10.1016/J.SCITOTENV.2022.154729.
- Ning Wang, Shiyue Zhao, and Sutong Wang. A novel clustering-based resampling with cost-sensitive boosting method to model and map wildfire susceptibility. *Reliability Engineering and System Safety*, 242:109742, 2 2024. ISSN 0951-8320. doi: 10.1016/J.RESS.2023.109742.
- James L. Tracy, Antonio Trabucco, A. Michelle Lawing, J. Tomasz Giermakowski, Maria Tchakerian, Gail M. Drus, and Robert N. Coulson. Random subset feature selection for ecological niche models of wildfire activity in western north america. *Ecological Modelling*, 383:52–68, 9 2018. ISSN 0304-3800. doi: 10.1016/J.ECOLMODEL.2018.05.019.
- Hamed Khosravi, Mohammad Reza Shafie, Ahmed Shoyeb Raihan, Md Asif Bin Syed, and Imtiaz Ahmed. Optimizing forest fire prediction: A comparative analysis of machine learning models through feature selection and time-stage evaluation, 12 2023. URL <https://www.preprints.org/manuscript/202312.0577/v1>.
- Jorge Pereira, Jérôme Mendes, Jorge S.S. Júnior, Carlos Viegas, and João Ruivo Paulo. Metaheuristic algorithms for calibration of two-dimensional wildfire spread

- prediction model. *Engineering Applications of Artificial Intelligence*, 136, 10 2024. ISSN 09521976. doi: 10.1016/j.engappai.2024.108928.
- Abolfazl Jaafari, Seyed Vahid Razavi Termeh, and Dieu Tien Bui. Genetic and firefly metaheuristic algorithms for an optimized neuro-fuzzy prediction modeling of wildfire probability. *Journal of Environmental Management*, 243:358–369, 8 2019. ISSN 10958630. doi: 10.1016/j.jenvman.2019.04.117.
- Jorge Pereira, Jerome Mendes, Jorge S.S. Junior, Carlos Viegas, and Joao Ruivo Paulo. Wildfire spread prediction model calibration using metaheuristic algorithms. In *IECON Proceedings (Industrial Electronics Conference)*, volume 2022-October. IEEE Computer Society, 2022. ISBN 9781665480253. doi: 10.1109/IECON49645.2022.9968435.
- Hao Zhang, Hui Liu, Guoqing Ma, Yang Zhang, Jinxia Yao, and Chao Gu. A wildfire occurrence risk model based on a back-propagation neural network-optimized genetic algorithm. *Frontiers in Energy Research*, 10, 1 2023. ISSN 2296598X. doi: 10.3389/fenrg.2022.1031762.
- Arip Syaripudin Nur, Yong Je Kim, and Chang-Wook Lee. Creation of wildfire susceptibility maps in plumas national forest using insar coherence, deep learning, and metaheuristic optimization approaches. *Remote Sensing*, 14(17), 2022. ISSN 2072-4292. doi: 10.3390/rs14174416. URL <https://www.mdpi.com/2072-4292/14/17/4416>.
- A’Kif Al-Fugara, Ali Nouh Mabdeh, Mohammad Ahmadlou, Hamid Reza Pourghasemi, Rida Al-Adamat, Biswajeet Pradhan, and Abdel Rahman Al-Shabeeb. Wildland fire susceptibility mapping using support vector regression and adaptive neuro-fuzzy inference system-based whale optimization algorithm and simulated annealing. *ISPRS International Journal of Geo-*

Information 2021, Vol. 10, Page 382, 10:382, 6 2021. ISSN 2220-9964. doi: 10.3390/IJGI10060382. URL <https://www.mdpi.com/2220-9964/10/6/382/html><https://www.mdpi.com/2220-9964/10/6/382>.

Mahdi Azizi, Uwe Aickelin, Hadi A. Khorshidi, and Milad Baghalzadeh Shishehgarkhaneh. Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Scientific Reports*, 13, 12 2023. ISSN 20452322. doi: 10.1038/s41598-022-27344-y.

Dieu Tien Bui, Quang Thanh Bui, Quoc Phi Nguyen, Biswajeet Pradhan, Haleh Nampak, and Phan Trong Trinh. A hybrid artificial intelligence approach using gis-based neural-fuzzy inference system and particle swarm optimization for forest fire susceptibility modeling at a tropical area. *Agricultural and Forest Meteorology*, 233:32–44, 2 2017. ISSN 01681923. doi: 10.1016/j.agrformet.2016.11.002.

Trang Thi Kieu Tran, Saeid Janizadeh, Sayed M. Bateni, Changhyun Jun, Dongkyun Kim, Clay Trauernicht, Fatemeh Rezaie, Thomas W. Giambelluca, and Mahdi Panahi. Improving the prediction of wildfire susceptibility on hawaii island, hawaii, using explainable hybrid machine learning models. *Journal of Environmental Management*, 351, 2 2024. ISSN 10958630. doi: 10.1016/j.jenvman.2023.119724.

Farhad Soleimanian Gharehchopogh, Isa Maleki, and Zahra Asheghi Dizaji. Chaotic vortex search algorithm: metaheuristic algorithm for feature selection. *Evolutionary Intelligence*, 15:1777–1808, 9 2022. ISSN 18645917. doi: 10.1007/S12065-021-00590-1/TABLES/17. URL <https://link.springer.com/article/10.1007/s12065-021-00590-1>.

A. N.K. Nasir, R. M.T. Raja Ismail, and M. O. Tokhi. Adaptive spiral dynamics metaheuristic algorithm for global optimisation with application to modelling of

- a flexible system. *Applied Mathematical Modelling*, 40:5442–5461, 5 2016. ISSN 0307904X. doi: 10.1016/j.apm.2016.01.002.
- Hongwei Ding, Yuting Liu, Zongshan Wang, Gushen Jin, Peng Hu, and Gaurav Dhiman. Adaptive guided equilibrium optimizer with spiral search mechanism to solve global optimization problems. *Biomimetics*, 8, 9 2023. ISSN 23137673. doi: 10.3390/biomimetics8050383.
- Madiyah Binti Omar, Kishore Bingi, B. Rajanarayan Prusty, and Rosdiazli Ibrahim. Recent advances and applications of spiral dynamics optimization algorithm: A review, 1 2022. ISSN 25043110.
- Lei Xie, Tong Han, Huan Zhou, Zhuo Ran Zhang, Bo Han, and Andi Tang. Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization, 2021. ISSN 16875273.
- NASA. Nasa — lance — firms, 2024. URL <https://firms.modaps.eosdis.nasa.gov/>.
- Era5 hourly data on single levels from 1940 to present, 2024. URL <https://cds.climate.copernicus.eu/datasets/reanalysis-era5-single-levels?tab=overview>.
- Bill Bell, Hans Hersbach, Adrian Simmons, Paul Berrisford, Per Dahlgren, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Raluca Radu, Dinand Schepers, Cornel Soci, Sebastien Villaume, Jean Raymond Bidlot, Leo Haimberger, Jack Woollen, Carlo Buontempo, and Jean Noël Thépaut. The era5 global reanalysis: Preliminary extension to 1950. *Quarterly Journal of the Royal Meteorological Society*, 147:4186–4227, 10 2021. ISSN 1477870X. doi: 10.1002/QJ.4174.
- Ioannis Prapas, Akanksha Ahuja, Spyros Kondylatos, Ilektra Karasante, Eleanna Panagiotou, Lazaro Alonso, Charalampos Davalas, Dimitrios Michail, Nuno

- Carvalhais, and Ioannis Papoutsis. Deep learning for global wildfire forecasting. 11 2022. URL <https://arxiv.org/abs/2211.00534v2>.
- Ritu Taneja, James Hilton, Luke Wallace, Karin Reinke, Simon Jones, Ritu Taneja, James Hilton, Luke Wallace, Karin Reinke, and Simon Jones. Effect of fuel spatial resolution on predictive wildfire models. *International Journal of Wildland Fire*, 30:776–789, 8 2021. ISSN 1448-5516. doi: 10.1071/WF20192. URL <https://www.publish.csiro.au/wf/WF20192>.
- John T. Abatzoglou and Crystal A. Kolden. Relationships between climate and macroscale area burned in the western united states. *International Journal of Wildland Fire*, 22:1003–1020, 2013. ISSN 10498001. doi: 10.1071/wf13019.
- John T. Abatzoglou and A. Park Williams. Impact of anthropogenic climate change on wildfire across western us forests. *PNAS*, 113:11770–11775, 10 2016. ISSN 10916490. doi: 10.1073/pnas.1607171113.
- John T. Abatzoglou, Crystal A. Kolden, A. Park Williams, James A. Lutz, and Alistair M.S. Smith. Climatic influences on interannual variability in regional burn severity across western us forests. *International Journal of Wildland Fire*, 26:269–275, 2017. ISSN 10498001. doi: 10.1071/wf16165.
- Naian Liu, Jiao Lei, Wei Gao, Haixiang Chen, and Xiaodong Xie. Combustion dynamics of large-scale wildfires. *Proceedings of the Combustion Institute*, 38: 157–198, 1 2021. ISSN 1540-7489. doi: 10.1016/J.PROCI.2020.11.006.
- E. Louise Loudermilk, Joseph J. O’Brien, Scott L. Goodrick, Rodman R. Linn, Nicholas S. Skowronski, and J. Kevin Hiers. Vegetation’s influence on fire behavior goes beyond just being fuel. *Fire Ecology*, 18:1–10, 12 2022. ISSN 19339747. doi: 10.1186/S42408-022-00132-9/FIGURES/2. URL <https://fireecology.springeropen.com/articles/10.1186/s42408-022-00132-9>.

Igor Drobyshev, Nina Ryzhkova, Jonathan Eden, Mara Kitenberga, Guilherme Pinto, Henrik Lindberg, Folmer Krikken, Maxim Yermokhin, Yves Bergeron, and Alexander Kryshen. Trends and patterns in annually burned forest areas and fire weather across the european boreal zone in the 20th and early 21st centuries. *Agricultural and Forest Meteorology*, 306:108467, 8 2021. ISSN 0168-1923. doi: 10.1016/J.AGRFORMET.2021.108467.

Stella Afolayan, Ademe Mekonnen, Brandi Gamelin, and Yuh Lang Lin. Multiscale interactions between local short- and long-term spatio-temporal mechanisms and their impact on california wildfire dynamics. *Fire 2024, Vol. 7, Page 247*, 7:247, 7 2024. ISSN 2571-6255. doi: 10.3390/FIRE7070247. URL <https://www.mdpi.com/2571-6255/7/7/247/html><https://www.mdpi.com/2571-6255/7/7/247>.

Farahnaz Fazel-Rastgar and Venkataraman Sivakumar. Weather pattern associated with climate change during canadian arctic wildfires: A case study in july 2019. *Remote Sensing Applications: Society and Environment*, 25:100698, 1 2022. ISSN 2352-9385. doi: 10.1016/J.RSASE.2022.100698.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 12 1997. ISSN 0004-3702. doi: 10.1016/S0004-3702(97)00043-X.

Naoual El Aboudi and Laila Benhlila. Review on wrapper feature selection approaches. *Proceedings - 2016 International Conference on Engineering and MIS, ICEMIS 2016*, 11 2016. doi: 10.1109/ICEMIS.2016.7745366.

Yi Zhao, Jiale Ma, Xiaohui Li, and Jie Zhang. Saliency detection and deep learning-based wildfire identification in uav imagery. *Sensors (Switzerland)*, 18, 3 2018. ISSN 14248220. doi: 10.3390/s18030712.

- Weiguo Zhao, Liying Wang, and Zhenxing Zhang. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems*, 163:283–304, 1 2019. ISSN 09507051. doi: 10.1016/j.knosys.2018.08.030.
- Mohd Herwan Sulaiman, Zuriani Mustaffa, Mohd Mawardi Saari, and Hamdan Daniyal. Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 87, 1 2020. ISSN 09521976. doi: 10.1016/j.engappai.2019.103330.
- Eva Trojovská and Mohammad Dehghani. A new human-based metahurestic optimization method based on mimicking cooking training. *Scientific Reports*, 12, 12 2022. ISSN 20452322. doi: 10.1038/s41598-022-19313-2.
- Afshin Faramarzi, Mohammad Heidarinejad, Brent Stephens, and Seyedali Mirjalili. Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191, 3 2020. ISSN 09507051. doi: 10.1016/J.KNOSYS.2019.105190.
- Mohamed Abdel-Basset, Doaa El-Shahat, Mohammed Jameel, and Mohamed Abouhawwash. Exponential distribution optimizer (edo): a novel math-inspired algorithm for global optimization and engineering problems. *Artificial Intelligence Review*, 56:9329–9400, 9 2023. ISSN 15737462. doi: 10.1007/S10462-023-10403-9/TABLES/29. URL <https://link-springer-com.ezproxy.tru.ca/article/10.1007/s10462-023-10403-9>.
- José Antonio Martín H., Javier De Lope, and Darío Maravall. Adaptation, anticipation and rationality in natural and artificial systems: Computational paradigms mimicking nature. *Natural Computing*, 8:757–775, 12 2009. ISSN 15677818. doi: 10.1007/S11047-008-9096-6.

- Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80: 8091–8126, 2 2021. ISSN 15737721. doi: 10.1007/S11042-020-10139-6.
- Prachi Agrawal, Hattan F. Abutarboush, Talari Ganesh, and Ali Wagdy Mohamed. Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access*, 9:26766–26791, 2021. ISSN 21693536. doi: 10.1109/ACCESS.2021.3056407.
- Olatunji O. Akinola, Absalom E. Ezugwu, Jeffrey O. Agushaka, Raed Abu Zitar, and Laith Abualigah. Multiclass feature selection with metaheuristic optimization algorithms: a review, 11 2022. ISSN 14333058.
- Kenichi Tamura and Keiichiro Yasuda. Spiral multipoint search for global optimization. In *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011*, volume 1, pages 470–475, 2011. ISBN 9780769546070. doi: 10.1109/ICMLA.2011.131.
- Andrey Polezhaev. Spirals, their types and peculiarities. *Frontiers Collection*, Part F1072:91–112, 2019. ISSN 21976619. doi: 10.1007/978-3-030-05798-5_4. URL https://www.researchgate.net/publication/332208865_Spirals_Their_Types_and_Peculiarities.
- Raph Levien. The euler spiral: a mathematical history. Technical report, EECS Department, University of California, Berkeley, Sep 2008. URL <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-111.html>.
- David H. Von Seggern. *Practical handbook of curve design and generation*. CRC Press, 1994. ISBN 978-0-8493-8916-0.
- Matteo Cherchi, Sami Ylinen, Mikko Harjanne, Markku Kapulainen, and Timo Aalto. Dramatic size reduction of waveguide bends on a micron-scale silicon

- photonic platform. *Optics Express*, 21:17814–17823, 7 2013. ISSN 10944087. doi: 10.1364/OE.21.017814. URL <http://arxiv.org/abs/1301.2197><http://dx.doi.org/10.1364/OE.21.017814>.
- Edward A. (Edward Albert) Bowser. An elementary treatise on analytic geometry, embracing plane geometry and an introduction to geometry of three dimensions , 1845-1910, 1910. URL <https://archive.org/details/anelementarytre09bowsgoog/page/n250/mode/2up>.
- Øyvind Hammer. *The Perfect Shape: Spiral Stories*. Springer International Publishing, 2016. doi: 10.1007/978-3-319-47373-4_15.
- Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968.
- Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937. doi: 10.1080/01621459.1937.10503522.
- Hao Dong, Han Wu, Pengfei Sun, and Yunhong Ding. Wildfire prediction model based on spatial and temporal characteristics: A case study of a wildfire in portugal’s montesinho natural park. *Sustainability*, 14(16), 2022. ISSN 2071-1050. doi: 10.3390/su141610107. URL <https://www.mdpi.com/2071-1050/14/16/10107>.
- Sally S.-C. Wang, Yun Qian, L. Ruby Leung, and Yang Zhang. Identifying key drivers of wildfires in the contiguous us using machine learning and game theory interpretation. *Earth’s Future*, 9(6):e2020EF001910, 2021. doi: <https://doi.org/10.1029/2020EF001910>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020EF001910>. e2020EF001910 2020EF001910.