2020

Some results on a set of data driven stochastic wildfire models

Maxfield E. Green
University of Vermont

Some results on a set of data driven stochastic wildfire models

A Thesis Presented

by

Maxfield E. Green

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Complex Systems and Data Science

May, 2020

Defense Date: March 17th, 2020
Dissertation Examination Committee:

Chris Danforth, Ph.D., Advisor
Safwan Washah, Ph.D., Chairperson
Peter Dodds, Ph.D.
Jarlath O'Neill, M.Sc.
Cynthia J. Forehand, Ph.D., Dean of Graduate College

Abstract

Across the globe, the frequency and size of wildfire events are increasing. Research focused on minimizing wildfire is critically needed to mitigate impending humanitarian and environmental crises. Real-time wildfire response is dependent on timely and accurate prediction of dynamic wildfire fronts. Current models used to inform decisions made by the U.S. Forest Service, such as Farsite, FlamMap and Behave do not incorporate modern remotely sensed wildfire records and are typically deterministic, making uncertainty calculations difficult. In this research, we tested two methods that combine artificial intelligence with remote sensing data. First, a stochastic cellular automata that learns algebraic expressions was fit to the spread of synthetic wildfire through symbolic regression. The validity of the genetic program was tested against synthetic spreading behavior driven by a balanced logistic model. We also tested a deep learning approach to wildfire fire perimeter prediction. Trained on a time-series of geolocated fire perimeters, atmospheric conditions, and satellite images, a deep convolutional neural network forecasts the evolution of the fire front in 24-hour intervals. The approach yielded several relevant high-level abstractions of input data such as NDVI vegetation indexes and produced promising initial results. These novel data-driven methods leveraged abundant and accessible remote sensing data, which are largely unused in industry level wildfire modeling. This work represents a step forward in wildfire modeling through a curated aggregation of satellite image spectral layers, historic wildfire perimeter maps, LiDAR, atmospheric conditions, and two novel simulation models. The results can be used to train and validate future wildfire models, and offer viable alternatives to current benchmark physics-based models used in industry.

in loving memory of

Roy C. Fields (June 1939 - January 2013)

Trinity McDonald (June 1996 - October 2016)

I want to contribute, to the chaos, I don't want to watch and then complain. At any rate, that is happiness; to be dissolved into something complete and great.
- B. Sella & W. Cather

Acknowledgements

Thank you to my parents, Kimber and Jordan Green for their endless love, support and astounding trust. Thank you to my sister Reese Joy-Fields Green: your intellect, dedication and authenticity are a frequent inspiration to me. Thank you to my primary advisor, Chris Danforth for frequently reassuring me that I in fact was on track to complete this project. Thank you to my colleagues and fellow students who contributed to this work immensely, specifically Todd DeLuca, Karl Kaiser, Nat Shenton and Dave Landay. Working on various components of this with you individually and in small groups has been a pleasure. I have learned so much from our conversations and late night white boarding sessions. Although it maybe always felt rushed and just barely put together, I think we are onto something here! Thanks for chasing last minute ideas with me, for testing weird progress tracking frameworks and for offering up your brilliance at all hours of the day! To my dear friend and valued collaborator Sophie Hodson, thank you for your edits, company and true belief in me over the years. I'm so thankful for our exceptional partnership. Our final pushes and general 11th-houring has brought much meaning to my time at UVM.

To my friends and classmates, its been a pleasure to rise (and fall sometimes) to many challenges with you. The faculty within and adjacent to the Vermont Complex Systems Center have given me a set of tools and a questioning lens that I am incredibly thankful for.

In addition to the many folks who have helped me on this project, I would like to nod to all the stone monkeys in the Valley, those who suffer from wildfires annually. This group of athletes inspired this work and push me daily to live my truth. Bradley, AB, Merryn, Jake and J, Thorn, Hannah, Carolina, J-Star, Molly, Colby, Dylan, "The

Belgians", HK and BG : Thanks for keeping the dream alive for me during moments of pause from climbing. Your commitment to trying hard is one of my greatest inspirations. Finding a balance is everything and I'm still looking.

To my excellent climbing partners who listen to me waxing about wildfire models and to my esteemed colleagues who listen to me waxing about climbing, thank you. It is not often that one feels nostalgic about a moment in real time as it is passing. Thank you to those who have taught me to push and push for what I think is right and to commit to what makes me feel most alive, this project being an example.

Table of Contents

List of Figures

List of Tables

Chapter 1

A Brief Overview of Forest Management and Wildfires with a Focus on North America

We describe the series of forest management polices that have likely resulted in the modern day 'megafires' burning through the Western half of North America throughout the dry season. We then detail the change in management that has lead to the importance of accurate and timely fire modeling techniques. Following, w examine and compare existing wildfire models. Finally, we propose two novel models that address this need, along with a curated data-set and open source API in which the models are trained on.

1.1 Forest and Wildfire Management Strategies in North America

A natural component of many healthy ecosystems, wildfires help to maintain a balanced carbon cycle. The burning of plant material cycles nutrients from aging plant populations to the forest floor, encouraging new growth. In many ecosystems, wildfires are healthy and a part of any warm dry season [2, 3]. However, in the past century, the United States has experienced an evolution of forest management policies, fundamentally altering the wildfire cycle. In the Eastern United States, wildfires have historically been small and relatively not harmful to wilderness urban intersections. This is due to a humid climate with ample precipitation throughout the year. However, during westward expansion in the early 1800s, large wildfires began to threaten European settlement [4]. Prior to European occupation of the great West, Indigenous Native American tribes had coexisted with natural wildfires in arid climates for thousands of years. In fact, in select settings, indigenous cultures were using controlled burns to manage prairies and forests to maintain agricultural practices. As Europeans began to migrate further west, sequestering indigenous lands, these practices of using fire in land management were lost [5]. In the late 1880s, the U.S. Army was the primary body responsible for responding to wildfires. The official policy was to suppress every fire immediately after detection, regardless of the ecology of the landscape. By 1905, the U.S. Forest Service was founded, and inherited the same Universal Suppression Policy. This aggressive tactic was incredibly effective at reducing the annual number of acres burned in North America. Between 1935

and 1960, the number of wild acres effected by wildfires dropped from $120,000km^2$ to $8,100km^2 - 20,000km^2$ annually [6]. However, many of the managed ecosystems benefited from seasonal burning. The absence of periodic burning as a result of the suppression policies established in the 1900s lead to unmatched tree growth. As trees mature, they naturally shed layers of bio-material, barks, branches, leaves, etc. These materials can build up on the forest floor. Over many seasons, the successive accumulation of bio-material can inhibit new plant life and act as tinderbox for potential wildfires. The material that collects underneath the tree canopy can aid in the growth of thick understory, a collection of moss and shrub like vegetation. The combination of decaying plant material and spreading groundcover vegetation can link together disconnected components of the forest floor, making it easier for fire to spread through the forest.

In the late 1900s, forest ecologists began to inform wildfire forest management policy, specifically the discovery that certain tree species actually required heat from fire to grow defensive barks, such as the towering Sequoia Trees of the east high country Sierra Nevadas. While the fire suppression policies reduced short term wildfire risk, it left many wildlands primed for massive wildfires due to the accumulation of over 30 years of dry and dead plant material. In the past 50 years, alternatives to strict fire suppression have made their way into forest management decision making. One alternative to a strict suppression based policy is to allow naturally ignited wildfires to "run their course". In response to negative effects of a strict suppression policy for many ecosystems, the Wilderness Act of 1964 [7] prompted forest managers to "preserve natural conditions" of forested areas. The Wilderness Act even referred to fire as a management tool, calling for the inclusion of controlled burns to maintain

and restore forests. The United States Forest Service released a guide to Wilderness Fire Use, stating that "fire, as a critical natural process, will be integrated into land and resource management plans and activities on a landscape scale and across administrative boundaries". In Figure 1.1 we see the fall and subsequent rise of yearly area effect by fire.

This change in strategy altered the ecology in many North American forests, thinning out bio material. However, many areas are still effected by strict burn policy and contain excessive flammable ground coverage [8].

In the current stage of wildfire management, we are still seeing less than historic annual acres burned, but the fires that do happen are tending to be larger, hotter and deadlier then ever before [9]. While there are fewer acres burned annually than in the past centuries (prior to European Colonization), due to suppression tactics, there are rising accounts of mega fires. Mega fires are wildfires characterised by their large size, duration, complex burn behaviors, cost to contain, and risk fatality to firefighters and community developments adjacent to wilderness areas [10].

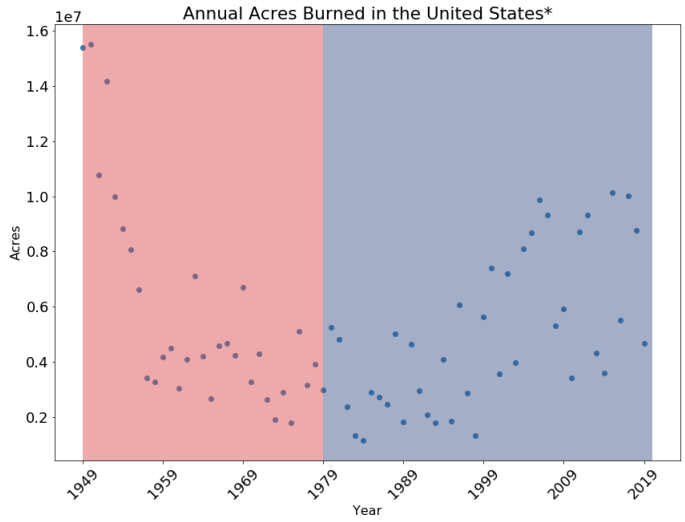


Figure 1.1: Early suppression tactics were incredibly effective at reducing the annual number of acres burned, however, we are still feeling the negative effects today. The red time period indicates fire suppression policy while the blue indicates data from post wilderness act. [1]

Current nationwide efforts to better understand the effects of forest management on wildfires are being lead by The Fire and Smoke Model Evaluation Experiment, FASMEE [11]. FASMEE is a large inter-agency organization that monitors artificial and naturally ignited wildfires. They study how fuels, fire behavior and meteorology interact to determine the dynamics of smoke plumes and vegetative response. FASMEE collects observations from large prescribed fires by combining Light Detection and Ranging (LiDAR), radar, ground monitoring, aircraft and satellite imagery, and weather and atmospheric measurements. Their mission is both one of scientific inquiry as well as public safety, to better predict fire behavior, the effects of fire on the landscape, and impacts of smoke. FASMEE represents a modern effort to leverage remote sensing technologies and a diverse team of scientists with backgrounds in ecology, forestry, climatology and applied mathematics.

1.2 Environmental, Societal and Economic Impact of Wildfire

Each year, between 4 - 8 million acres of land are damaged by wildfires. Just in the past 10 years, this has represented a $5.1 billion cost in infrastructural damage repair [12].

Although there has been a clear increase in annual acres burned, this does not directly translate to negative ecological impact. In some cases, a burned landscape will act as a fire break in the next season, separating once connected components of the forest. Future fire seasons are effected by fuel regrowth, human interference, topography and a changing climate.

Wildfires have an distinct effect on local labor markets during fire season. Being one of the only natural disasters that humans try to manage, or perhaps fight, communities see a small immediate bump in wages and earnings during a fire event. This is followed by an equivalent depression. It has been observed that larger and more frequent fires amplify this seasonal oscillation in temporary economic gain [13]. Allocating fire suppression to local agencies can help balance the economic hit from infrastructure loss. According to a case study of fire events in Trinity, California, local employment increased 1 percent during the first quarter of the fire for every 1 million spent in the county. Additionally, on average, 9% of suppression funding from the Forest Service is spent in the county in which the wildfire occurred. Contracts for suppression and support services are a central avenue for local capture. However, local business capacity appears to limit the ability of rural and resource-dependent

counties to capture suppression contracts.

Building predictive models to aid in wildfire preparation and containment efforts is increasingly important. With the advance in computational resources, wildfire modeling has become a key component to successfully forest management. Accurate simulations of wildfires can inform best practices for forest management, as well as real time response to wildfire events. In the past ten years, wildfire modeling has grown from fully physical models to data driven models that leverage artificial intelligence and increased coverage of fire events.

Traditional physical models are derived from the fundamental laws of physics and chemistry. They model coupled dynamics of the physical systems like diffusion, advection, radiation, etc. These dynamics are often described with sets of coupled partial differential equations to compute how the system will evolve over time. In this literature review, we will describe a select number of the benchmark physical models that are currently used to guide forest management. Additionally, we will discuss the limitations of these models and introduce some modern works that have inspired the direction of this thesis.

Chapter 2

Modeling Wildfires

2.1 Modeling Fire and Wildfires

The spread of fire over a landscape includes processes and mechanisms that act at different physical and temporal scales. Energy is released in the form of heat due to chemical reactions during combustion. The energy is transferred to nearby fuel sources (vegetation) furthering material ignition. Modeling such complexity at varying temporal and spatial scales is not a trivial problem. The methods, assumptions and results that support wildfire models are still heavily disputed and can even be contradictory. As the impacts of wildfires are growing, and the complex dynamics of wildfires are still being discovered, the treatment of wildfires as a complex multiscale dynamical system is incredibly important.

We will briefly summarize the chemistry and physics that fundamental physical models of fire spread are based upon. Then, we will show how these fundamentals are applied in several different physical models, commenting on known limitations, assumptions and model validation. A review of these methods helped inform and

shape the direction of the original models presented in this thesis.

In their 1997 review, Grishin et. al [14] suggested four steps in the progression of modeling a wild fire. In this time, models were primarily based on derivations of physical systems in the form of coupled partial differential equations. Grishin's four steps are paraphrased below :

1. Instrument a physical analysis of wildfire spread by isolating the mechanism that controls transfer of energy from the fire perimeter to adjacent entities.

2. Define a medium and then determine its reaction and thermophysical properties. Derive a set of equations to satisfy these conditions.

3. Solve the set of equations using the best possible numerical methods.

4. Evaluate the accuracy of the model by comparing it to the real system.

In modeling many dynamical systems, one must make simplifying assumptions in order to make the problem tractable and feasible to parameterize. Grishin's four steps are a general, albeit lofty, guide to modeling many physical systems. One component that Grishin leaves out that is a particular challenge in physical modelling is capturing edge cases.

Fire and Combustion

Many of the fire spreading models used by foresters and other practitioners are expansions upon Rothermel's [15] one dimensional spread model that predicts the surface and crown rates of fire spread [16]. Rothermel's equation form is given in Fig. 2.1.

$$R = \frac{I_R \xi (1 + \phi_w + \phi_s)}{\rho_b \varepsilon Q_{ig}}$$

Figure 2.1: Rothermal's Equation for Surface Spread Rate

Two dimensional models combine the work of Rothermel with Huygen's principle of propagating waves [17]. Such fundamentals are still present in practice in models such as Farsite [18], Firetec [19] and FlameMap [20]. Full simulations iterate Rothermal's equation over time to calculate fire acceleration.

Additionally, there are models that have received much praise from the wildfire research community, but have not yet been implemented in forest management [21], [22], [23]. Before examining a set of applied and well-known academic models, we will review some of the treatments of the physical processes that lead to combustion from an elementary physics perspective. We will paraphrase the treatment of typical combustion reaction modeling found in the comprehensive review by Sullivan et al [24] recounting modelling from 1970-2007.

The fuel sources of a wildfire are diverse and vary spatially. Thus there is a large range of chemical compounds and biological fuel sources which alter the speed of ignition and the spread of a fire across a landscape. Modern complete physical models account for processes within the stratum of vegetation, and thus do require a high resolution of fuel type. The treatment of vegetation stratum dynamics is discussed.

The primary chemical components of wood fuel are cellulose, lignin, and hemicellulose. The combustion and energy transfer from biomass is usually treated as the sum of the fuel's main components [25]. Additional inorganic matter inside the vegetation can both inhibit and aid the combustion process.

The process of burning will only occur after vegetation has been dried. Specifically, when the cellulose is heated up, the cell undergoes thermal degradation in one of two forms. In the absence of oxygen, the vegetation will dry up and produce *tar* substance as a result of the chemical reaction. If sufficient heat is present, the tar can combust. The secondary possibility results in the formation of *char* and occurs in the presence of moisture and low heat. These two processes compete with one another [24] but at a threshold, the system will converge and support one.

The quantity of energy that determines whether or not a chemical reaction will take place is referred to as the activation energy E_a. This is the amount of energy required to catalyze a reaction. This in turn controls the rate of the reaction according to the Arrhenius law

$$k = A^{\frac{-E_g}{RT}} \tag{2.1}$$

where k is the reaction rate constant, A is a pre-exponential factor, R is the gas constant and T is the absolute temperature of the reactants.

The physical processes that produce heat transfer are primarily radiation and advection [[14], [26]]. In the presence of winds, advection drives the spread of the fire, while alternatively, fire spreads primarily from radiation. Radiant heat can be simplified to the Stefan-Boltzmann radiant heat transfer equation (RTE) [24].

$$q = \sigma T^4 \tag{2.2}$$

Where σ is the Stefan - Boltzmann constant and T is the temperature of the radiating surface (K) [27]. However, recently there have been more complex adaptations to the RTE treatment problem [28].

The processes of advection and fluid transport are typically handled through the conservation of mass and momentum equations. We can write the change in density over time of a fluid as a function of the fluids density and velocity.

2.1.1 Model Types and Review

There are many different scales in which to model wildfire. There is active research in both short and long time scales. The dynamics of a flame can be examined over the course of fractions of a micro-second [29], all the way up to 10,000 year long simulations. Long term models hope to predict the effects of multiple millennia of fire seasons [30]. This approach can help us understand the threshold at which fire is beneficial in growing healthy forests, and can act as prevention for future natural disasters. The primary interest of models used in the field lie in the day to week range, as they are typically used in response to active wildfire spreading. However, micro and macro temporal models are very useful in understanding additional underlying dynamical properties of the system. Additionally, fire is modeled at varying spatial resolution, altering the relevant fuel components. At a very high spatial resolution, we are interested in materials down the vegetal stratum as discussed in the combustion

section. However, when modeling a large section of a forest, this fine-grained focus may not be feasible. In this case, we are interested in approximating the effect of micro structures in combustion behavior by coarse graining to the surface vegetation type level, such as the species of tree or bush, or perhaps the distribution of crown and canopy height and resulting density. Now we see that this system is incredibly complex, as there is an incredible spectrum of scales to consider examination within. An additional distinction in modeling wildfire events is the treatment of the continuity of space. Systems modeled in continuous space are described using coupled sets of partial differential equations [31], [23], [32], [33], [34] Alternatively, models propagate a fire front over discrete space. Both strategies are discussed in this review.

Table 2.2 highlights the primary models that are in development or current use by wildfire fighting agencies to help predict evolving fire fronts. There are roughly three classes of model present in this review, empirical, semi empirical and physical. To begin this review of methods, we will examine models at varying scales of focus in time and space. The focus of this review will be on short term models that predict the spreading of the fire perimeter at the day to week resolution.

Table 2.1: Model Review Reference Table

Model	Description	Year	Reference
Farsite	Semi-Emprical Fire surface spreading simulation based on Huygen's Law applied to expanding polygons along the fire perimeter	1998 - Present	[18]
FlamMap	Geo spatial model that predicts time invariant fire behaviors.	1999	[20]
MultiScale	A complete physical model that treats combustion at three different interacting spatial scales. Not validated at this time.	2014	[23]
WFR-SFIRE	Weather Research and Forcasting - Spread Fire. Coupled atmospheric and surface fire spreading model	2011	[22], [21]
Data Assimilation	This was a precursor to WFR - SFIRE that allows practitioners to update input data in real time as the model is forecasting.	2009	[21]
WFDS	Wildland - Urban - Interface Fire Dynamics Simulation. Computational fluid dynamics to resolve buoyant flow, heat transfer, combustion, and thermal fuel depletion.	2010	[35]
BehavePlus	Surface and crown fire spread, fire source dependence, containment strategy suggestion,tree death	2007	[36]

Table 2.2: Relevant wildfire forecasting models under use to historically used by the U.S. Forest Service. Reported year indicates the year the initial paper introducing the model was published. However, many models have been since initial publication.

2.1.2 FlamMap and FARSITE

FlamMap is a well-known spreading model implemented in 2006 that is used in many wildfire fighting communities in North America. The model is used to analyze spatial variability in fire behavior and supports three different output sets. The necessary inputs are mapped into the model using geographic information systems (GIS). The landscape in which the fire spreads is discretized over a 2D grid. All fire calculations assume that fuel moisture and wind velocity are constant in time. The model considers the elevation, slope, fuel model, canopy cover, canopy height, crown base height and crown bulk density. Each input variable is stored in a matrix where a given location represents a cell on the landscape grid. Figure 2.2 is provided as a sample landscape from the FlamMap documentation [20].

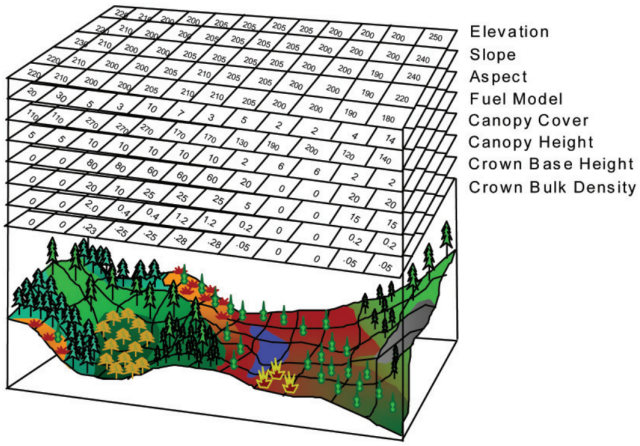


Figure 2.2: An example FlamMap input data structure composed of several matrix layers containing numeric values representing geospatial environmental conditions such as elevation and aspect of the landscape topography. In addition to the pictured variables, winds are either entered into the system as a fixed speed and direction for the entire landscape, or as a spatial wind field, represented as two grids, one for speed and one for direction [1].

The first feature of FlamMap produces a set of basic wildfire behavior descriptions

based on a constant set of environmental conditions for an entire landscape. The output of this model feature are a set of rasters that provide fire behavior on the grid cellular level. These layers include Fireline Intensity, Flame Length, Heat per unit Area and Midflame Windspeed. Each of these layers is for a single time, as there is no temporal aspect within FlamMap. Additionally, FlamMap is able to calculate fire growth in the absence of any time varying environmental conditions. The path of fire spread is calculated according to the the "Minimum Travel Time" algorithm [1].

Minimal Travel Time (MMT)

In an unweighted and undirected network, the minimum travel time between two nodes is the number of edges that connect the nodes. In large networks, there will be multiple paths that connect two nodes. However, when using a grid to model land area, the nodes are regularly spaced and a fire event at one cell in the grid does not imply burning or occupying all of the associated land area. In MMT, fire is able to travel from cell corners. The fire boundary grows to maintain an elliptic shape according to Huygen's principle. Huygen's principle assumes that each ellipse will spread synchronously and independent from one another [17]. The ellipse's spread rate from a point source in the X and Y direction are given by Equation 2.4

$$\frac{\partial X}{\partial t} = a \sin(\theta) \tag{2.3}$$

$$\frac{\partial Y}{\partial t} = b \cos(\theta) + c \tag{2.4}$$

Each cell will exhibit different fire spread behavior according to local slope, fuel types and wind behavior. The MMT algorithm returns the minimal time and path

that the simulated fire will spread between a location of interest and the current position of the fire. This data is output in the form of estimated arrival time contours. MMT expands the fire front based on an application of Huygen's law which will expand the perimeter according to independent wavelets as described in Figure 2.3.

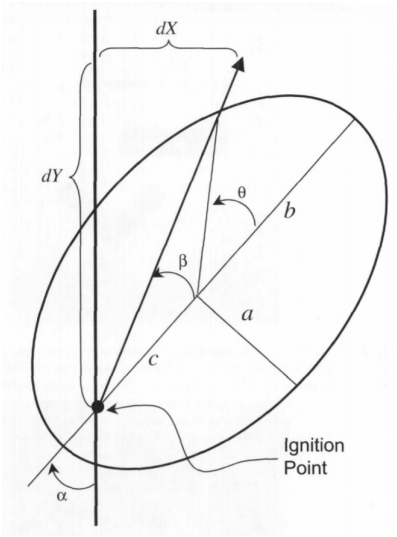


Figure 2.3: Elliptical fire spread rate dimensions(a,b,c) and angles (α, β, γ) for computing fire travel times from the ignition point over an arbitrary distance defined by dX and dY. Huygen's law will produce the X_t and Y_t, the spreading rates of the ellipse's semi major and minor axis [1].

The final component of the FlamMap system is a fuel treatment optimization engine. This recalculates fire behavior by changing the fuel model based on the major MTT pathways. This can help inform wildfire fighters in how to optimally place fire breaks in a landscape given ensambled predicted trajectories of the fire. These three features of FlamMap can help decision makers optimally treat fuel hazard and manage oncoming fires. Alone, this model is not enough to show how a fire shape is most likely to grow or change over time. It is typically joined with FARSITE, an alternative method that is a full fire simulation system [18].

The FARSITE model takes the same input parameters as FlamMap but calculates the fire behavior characterizations over time, producing a real time simulation of a wildfire evolution over a landscape.

Limitations of FARSITE and FlamMap

The FARSITE simulations are based on a set of coupled mathematical models that approximate the relationship between fuel, topology, weather and fire behavior. In the simplest cases, the system has produced very accurate results, given smooth landscapes and atmospheric conditions [18, 20]. Uniform surface fire spreading shape assumption based models are able to reproduce behavior exhibited by real observed fires. However, in non uniform terrain or during extreme weather events, these assumptions start to wear on model performance. Validation studies [37] have shown that FARSITE struggles to produce accurate results on extreme fire events such as plume-dominated fires, fire whirls and mass fires.

Additionally, it is assumed that the spread of fire from one point to another occurs independently from the dynamics of surrounding points of propagation. I.e. the shape and length of a fire front is assumed to have no effect on the fire behavior at a given point. However, it has been observed that fires frequently generate their own local atmospheres, creating wind currents and even fire storm clouds that can produce lightning [38]. This fascinating positive feedback loop directly impacts observed spreading patterns.

2.1.3 Physical Models

Fully physical models are based on the balance of energy and mass during reaction-diffusion, radiation and advection. The form of such models is a set of coupled partial differential equations (PDEs). This review will focus on newer models from the age of large scale numerical methods. While the models presented are not currently used by the Forest Service to aid in fire prediction, they are well thought of in the research community. A limitation of some academic models such as the work done by [32] is that the derivations are not implementable and theoretic due to constraints on PDE stability.

In a simplified model, a set of 2D reaction - diffusion equations are used to describe the main spreading and burning processes. However, there are also "complete physical models" which attempt to accurately model interaction between the atmosphere and the fire environment, and specifically, how the fire can alter its own atmosphere [[16], [23], [32]]. These models are often not feasible to be simulated. Margerite et al [32] proposed one of the first complete physical models which has since been simplified to two dimensions and simulated computationally.

Sero - Guillaume Multi Scale Model

Sero-GullaumeO et al [23] proposed a multi-scale model that considers 2 phases, solid and gas, in three regions of the forest- above, below and inside at three different scales: microscopic(plant cell solid/gas), mesoscopic(twig, leaf and branch) and macroscopic(forest canopy and atmosphere). 2.4 is pulled from their 2001 publication to describe the interaction between these scales. This model simplifies combustion

by limiting it to the gas phase. Phase transitions from solid to gas are modeled with interface jump relations. Conservation of mass, momentum and energy are calculated from the mesoscopic solid phase and gas phase interaction. To describe events at the macro scale, mesoscale properties are averaged by convoluting the mesoscopic equations to a macroscopic scale.

The states of different fuel compounds inform processes at the mesoscopic scale. The mesoscopic scale models the drying, decomposition and eventual combustion of wild vegetation. They consider a generic vegetation covering composed of hemicellulose, cellulose and lignin with liquid and gas components of water and air respectively. When the area heats up, the vegetation begins to dry and generates a flammable gas.

The study derives a system of equations for the different phases of the vegetation components as well as a set of "jump conditions" between the solid vegetation phase and the flammable gaseous phase. Further, the proposed macroscopic scale models a collection of mesoscopic reactions happening in coordination with one another.

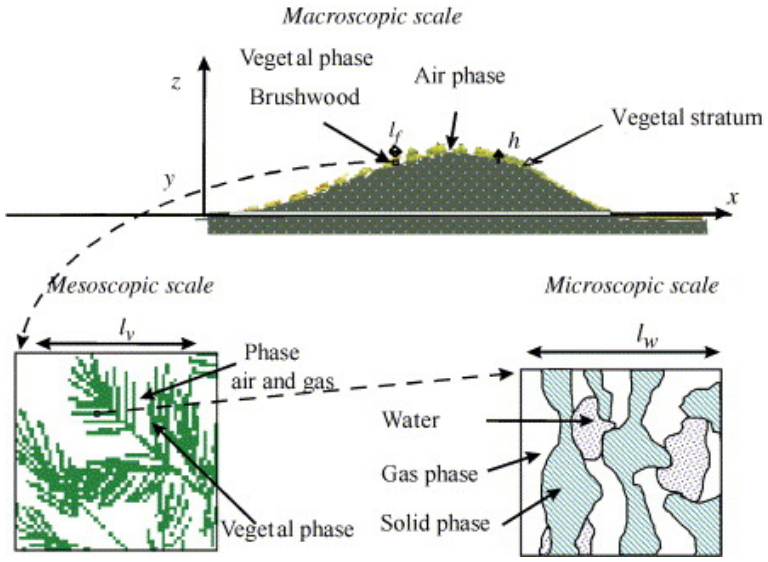


Figure 2.4: Simulation Scale Interaction

The product of this model is a large system of equations derived from the conservation of energy, momentum and mass at each of the scales. Simplifying the system under some relaxations produces the following system of equations for each phase of the particle.

$$(1-\Phi)\rho_p C_P^p \frac{\partial T_p}{\partial t} + \nabla \cdot (-\lambda_p \nabla T_p + Q_{pr}) = R_{pc} - \chi(T_f - T_p)$$
$$\Phi\rho_f C_P^f (\frac{\partial T_f}{\partial t} + V_f \cdot \nabla T_f) + \nabla \cdot (-\lambda_f \nabla T_f + Q_{fr}) = R_{pc} + \chi(T_f - T_p)$$

See [23] for full model derivation. We describe the simplified version used in simulation below as it is the one implemented in simulation.

Guillaume - Margerit Simplification and Simulation

Guillaume and Margerit propose a simplified 2 dimensional version of the theoretical model proposed in their follow up publication [32].

The group produces a simulation that considers vegetation, slope of terrain and wind as independent input variables. Drying and combustion of vegetation by reaction - diffusion are considered mechanisms. This model is a bridge between two dimensional reaction diffusion models and complete physical models. They propose a bound on the size of the fire (L) given the height of the vegetation(δ), such that

$$\frac{\delta}{L} = \epsilon \tag{2.5}$$

is a small parameter. With this assumption held, then the vegetation is a boundary layer between the fire and the surface.

The reduced model is derived by performing a matched asymptotic expansion of

the process inside the vegetal stratum and an accompanying outer expansion for those outside. The expansion is performed on the set of ordinary differential equations that balance energy, mass and momentum. The bounded *epsilon* is then used to expand all relevant quantities as :

$$f = f_0(x, y; t; z_1, t_1) + \epsilon f_1(x, y, t; z_1, t_1) + \epsilon^2 f_2(x, y, t; z_1, t_1) + ... \tag{2.6}$$

The expansion results in the following set of equations.

$$(1 - \Phi)\rho_p C_p^p \frac{\partial T_p}{\partial t} = \lambda_p \delta_s T_p - \nabla_s \cdot Q_{rp} + R_{cp} + \chi'(T_f - T_p) + M_r \tag{2.7}$$

$$\frac{\partial}{\partial t}((1 - \Phi)(1 - E^p)Sa_{wp}) = -(1 - \Phi)(1 - E^p)Sa_{wp}k_{wp}(T_p) \tag{2.8}$$

$$\frac{\partial}{\partial t}((1 - \Phi)(1 - E^p)Sa_{cp}\rho_c p) = (1 - \Phi)(1 - E^p)Sa_{wp}k_{w_p c_p}\rho_{w_p}(T_p) \tag{2.9}$$

$$\frac{\partial}{\partial t}((1 - \Phi)(E^p Sa_{lp})) = -(1 - \Phi)E^p Sa_{lp}k_{l_p v_{gp}}(T_p) \tag{2.10}$$

$$*with \tag{2.11}$$

$$C_P^p = (1 - E^p)(Sa_w \frac{\rho_{w_p}}{\rho_p} C_p) \tag{2.12}$$

However, this model is further reduced for simulation. In implementing the model, the authors neglect the difference between M_r and Q_r, the internal radiation heat source and the external. They progress the system of equations through time using the explicit Euler method. The main contribution of this model is the treatment of the radiative heat flux. In the supposed complete model, not simulated in the publication, the radiative heat flux is evaluated as a convolution integral. The authors note that the implemented model does not reflect a fire's behaviors due to this limitation but offer the full model derivation in the paper [32]. No validation is provided.

Mandel et al, Data Assimilation

This paper considers a simple 2 equation model with the goal of assimilating real time environmental data into the model as a means of course correction. This way, as the model is running, new data can be added to increase the accuracy of the predictions. Specifically, Dynamic Data Driven Application (DDDA) System techniques were employed. This allows the program to add and subtract real time data. The current model solves a system of reaction - convection - diffusion equations. The reaction rates are determined by the Arrhenius Equation developed by Svante Arrhenius in 1889 [39]. This equation $k = Ae^{\frac{-E_a}{RT}}$ shows the relationship between change in temperature and a reaction rate as several chemical species interact with one another. The model is derived from the conservation of energy, balance of fuel supply and the fuel reaction rate:

$$\frac{dT}{dt} = \nabla(k\nabla T) - \vec{v}\nabla T + A(Se^{\frac{B}{(T-T_a)}} - C(T - T_a))$$
$$\frac{dS}{dt} = -C_s S_s^{\frac{-B}{(T-T_a)}}, T > T_a;$$

The DDDA allows for the coefficients of the model equations to by dynamically updated to account for the new data.

To break down the set of equations we define each of the terms.

1. $\nabla\dot{(}\nabla T)$ models the short range transfer of heat due to radiation
2. $\vec{v}\dot{\nabla}T$ represents heat redistribution from wind gusts (advection)
3. The term $Se^{\frac{B}{(T-T_a)}}$ represents the rate at which fuel is being burned.

4. The $AC(T - T_a)$ models heat lost to the atmosphere through convection.

5. Finally, the reaction rate term is adjusted from the Arrhenius law [39] in the exponent to force zero reaction at ambient temperature. Giving way to $e^{\frac{B}{(T-T_a)}}$

The derivation of the system of PDEs is described below. The system is based on the fundamental conservation of energy and fuel reaction rate. The chemical reactions that take place from fuel combustion releases heat. The transfer of heat is due to the radiation and convection to the atmosphere. Short range heat transfer is modeled by diffusion. The two dimension heat flux through a segment per unit length is given by :

$$\vec{q_r} = -k_i \nabla T (W m^{-1}) \tag{2.13}$$

As the rate of the reaction is only dependent on the temperature, the rate at which the fuel is lost is proportional to the rate of reaction and the amount of fuel available. The rate of reaction is given by $C_s r(T)$ where C_s is a coefficient of proportionality and r is dimensionless.

$$\frac{dF}{dt} = -FC_s r(T) \tag{2.14}$$

Then the heat generated per unit area must be proportional to the amount of fuel that is lost

$$q_g = A_1 FC_s r(T), (W m^{-2}) \tag{2.15}$$

Other studies have examined special cases of the model presented. Weber et al [26] completes a formal expansion of the Arrhenius reaction model to determine the wave speed and predict small fire's spreading capability. [40] Another study predicts the wave speed upon fuel ignition and fire extinction. The stability of combustion waves are analyzed using asymptotic expansion by [41].

2.1.4 Mandel et al 2011, Coupled WRF - SFIRE

Mandel et al, [22] propose the Weather Research Forecasting and Surface Fire Spread (WFR-SFIRE) model. The WRF-SFIRE model accounts for the effect that the fire has on the local atmospheric and the resulting positive feedback loop. High winds contribute greatly to a growing fire and in turn the heat output from the fire causes air to rise rapidly to balance densities, this causes additional air to be pulled into the the empty space from nearby introducing a new current. The WFR - SFIRE model hopes to capture these coupled dynamics between the atmosphere and the fire.

The physical model consists of functions that calculate the rate of fire spread and offset heat fluxes. There are 13 different fuel type categories that the model can handle based on Burgan's [42] fuel index categories .

The fire spread rate is given by the modified Rothermel formula

$$S = R_0(1 + \phi_W + \phi_S) \tag{2.16}$$

Where R_0 is the rate prior to wind and ϕ_W is the effect of wind and ϕ_S is the effect of slope. As the fire spreads over a landscape, the fuel at a given position $F(t)$ decays exponentially over time. The fire is propagated by integrating the partial differential

level set equation. The integration is handled by Heun's method, a second order Runga- Kutta method.

Chapter 3

Case Study 1: Genetic Programming, Cellular Automaton and Synthetic Data

3.1 Modeling spatial spreading process with statistical learning

To reduce computation time, increase accuracy and leverage the advances in satellite imagery, recent work has modeled wildfire dynamics with machine learning or evolutionary strategies. This area has seen great success with increased accuracy of perimeter prediction from historic fires [43], [33]. Crowley et al [34] applied a set of reinforcement learning algorithms to learn spreading policies from satellite images within an agent based model.

Radke et al proposed a deep neural network algorithm titled FireCast that pre-

dicts 24 hour wildfire perimeter evolution based on Satellite images and local historic weather [33]. FireCast achieves a 20% higher average accuracy compared to the Farsite model [18] used in current practice.

Spatial spreading processes are commonly simulated using cellular autamaton (CA) [44], [45], [43] and agent based models ABMS [46], [47]. ABMs can been used to simulate complex systems by prescribing rule sets to independent agents. In the case of a wildfire model, each cell on fire represents an agent that can spread across the landscape and ignite neighboring cells based on a probability distribution that considers information about the current neighborhood. By evolving the function that governs agent behavior, agent based models can be used to predict system level spreading based on ground truth data. System level behavior is predicted by fine tuning agent decision functions. For example, Zhong et al [48] modeled evacuation crowd dynamics by evolving the agent rule set. This work aimed to predict the decision making process of an individual in an emergency evacuation of a building. Agents choose which exit to leave a building from based on distance, probable safety and volume of other agents headed that way. An optimal rule set will balance each of those variables to optimize the likelihood that all agents are able to leave the building safely. Fitting a symbolic regression to simulation results using an evolving rule-set exposes a population probability distributions that optimally weigh the considered variables. A number of fields have used this method to build realistic simulations used for further system prediction [49], [50], [51].

We propose a CA that trains a series of genetic programs to replicate seen and unseen wildfire simulations. We first introduce the mechanics of the CA, then give an overview of the evolutionary process and experimental design. We show that the

underlying spreading behavior can be learned and replicated by a genetic program based on synthetic environmental features.

3.2 Methods

We fit a symbolic regression to data generated by wildfire simulations. Agents represent instances of fire that spread according to a function of their local Von Neumann neighborhood [52]. We propose a naive spreading function and examine how well the regression can reproduce the spread patterns generated. We compare different genetic programs embedded in a CA by calculating how well they reproduce synthetic burns. We will first describe the CA that simulates the spread of fire and the generation of the synthetic data, then we will discuss the different evolutionary algorithms that attempt to learn the rules that govern the spread of fire.

3.2.1 Cellular Automaton

CAs were initially proposed by Von Neumann and are used to model spreading dynamics in discrete time and space [53]. CAs are well suited for simulating spreading on a grid. Each site on the grid has attributes that describe its unique state. The behavior of interest spreads across the grid when cells adapt according to their neighbors. In modeling a wild fire, we are interested in the relations between ignition probability and a number of local environmental factors, such as wind speed and direction, temperature, and relative humidity. Over time fire spreads from one cell to another based on a probability distribution that treats these factors as (learnable) parameters.

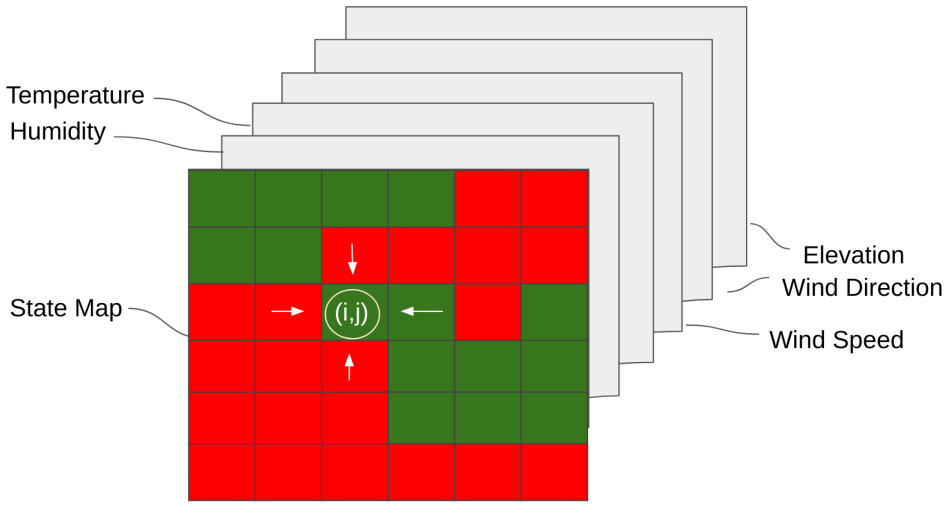


Figure 3.1: At each time-step of the simulation, all cells that are adjacent to a cell on fire are considered for ignition. The probability of ignition is determined by passing features from the cites Von Neumann neighborhood into an ignition probability distribution. Features are calculated from the 6 layers present in the visualized data structure, temperature, humidity, state, wind direction, wind speed and elevation.

As we show in `Fig.3.1`, each position on the landscape grid has six attributes: elevation, wind direction, wind speed, temperature, humidity, and burn state. The burn state attribute represents the state of the cell, in this case either on fire or not. The Von Neumann neighborhood is defined as the four orthogonal cells as indicated by white arrows in Fig. 3.1. Spreading behaviors on a grid surface are often modeled using this type of neighborhood [52]. At each time-step, the attributes of the neighborhood of each cell that borders the fire front contribute to the probability that the cell will catch fire.

In traditional CA models, the probability distribution that determines if a cell will adapt the behavior of its neighbor is static and prescribed by the CA designer. However, in this work, we evolve the spreading function through symbolic regression.

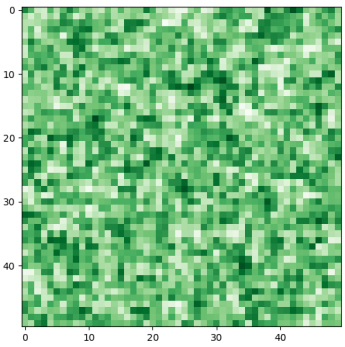
In each simulation, the CA runs for t time-steps and each cell updates its state based on its neighbors' attributes. At the end of a time-step, the perimeter of the fire expands probabilistically. At the end of the simulation, we retain an array of the coordinates of ignited cells.

Synthetic data generation

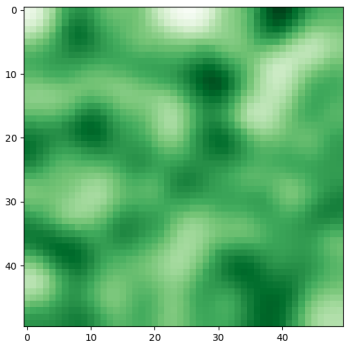
We simulate fires that spread over landscapes made of the six layers described in 3.1. The state layer is a binary matrix indicating whether a site is on fire or not. To build all other layers we generate matrices of smoothed-random floating points by implementing the Perlin noise algorithm [54] [55]. This iterative technique allows for the user to control how smooth or rough the generated spatial distribution is.

`Fig.3.2a` - `Fig.3.2d` shows the evolution of a Gaussian sample from random noise to a smoothed landscape. Over many iterations, the random sample begins to resemble a realistic smooth landscape, as seen in `Fig. 3.2`, which is the result of 100 iterations of this procedure.

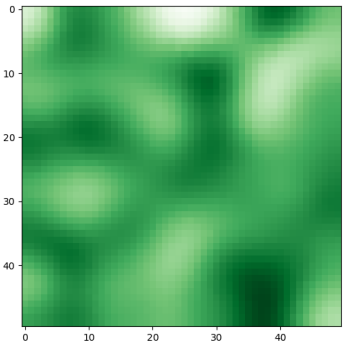
We generated each layer of a landscape using Perlin noise due to its abilities to produce natural gradients and its longstanding use in computer generated images [56,57]. We generated a single layer for each attribute considered in the model (topography, wind speed and direction, etc). We define a binary state layer that changes over the course of a simulation. The binary state layer encodes the state of every cell at a given time as either on fire (1) or not (0). These matrices are then stacked together to form a landscape in which a fire can be simulated on.



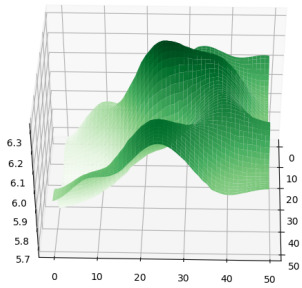
(a) Random noise, t_0



(b) Perlin Noise, t_{24}



(c) Perlin noise, t_{49}



(d) Perlin noise, t_{99}

Figure 3.2: Random values are sampled from Gaussian distributions and embedded over a matrix. We interpolate slopes between neighboring sites within the matrix, and their neighbors to describe smooth gradients over the sampled values.We use the quintic fade function $6t^5 - 15t^4 + 10t^3$ to interpolate smooth curves between all sites and those within their neighborhood

Feature engineering

Using the six attributes layers – elevation (Z), wind direction (wd), wind speed (ws), humidity, and burn states (S), we generate a set of four features that describe how the neighborhood affects the ignition likelihood of a central cell. In the case of the temperature and humidity, we take the average over the neighborhood, producing

features θ, ϕ, respectively. Since fire is more likely to spread uphill then down, the elevation feature, γ, weighs fire that is downhill from the central cell more than fire that is uphill from the central cell as fire moves uphill more quickly as reported by the Verisk Wildfire Analysis group, [12]. Thus, we have

$$\gamma = \sum_i I(S_i = 1) e^{-\Delta Z_i} \tag{3.1}$$

$$\Delta Z_i = Z_i - Z_0, \tag{3.2}$$

where Z_i is the elevation of the i-th neighbor or the central cell, Z_0 is the elevation of the central cell, and $I(S_i = 1)$ is an indicator variable that is 1 when the i-th neighbor is on fire and 0 otherwise.

Akin to elevation, we define a wind feature that reflects the fact that fire spreads downwind more readily than upwind. The wind feature, ω weighs fire that is upwind more than fire that is downwind. The wind feature is given in equations `Eq.3.3`, `Eq.3.4`, where ws represents wind speed, wd represents wind direction with i and j represent the spatial coordinates of the cell on the grid.

$$\omega = \sum_i I(S_i = 1) e^{w_i} \tag{3.3}$$

$$w_i = ws_i * (cf_i * \cos(wd_i) + sf_i * \sin(wd_i)) \tag{3.4}$$

where cf_i is a horizontal factor and sf_i is a vertical factor, corresponding to cosine and sine evaluation. Both factors are based on the relative position of the i-th neighbor to the central cell and are used to include the component of the wind that is

blowing toward (or away from) the center cell. For example, if the neighbor is north of the central cell, then $cf_i = 0$ and $sf_i = -1$.

These features then become inputs to a probability distribution that determines if a given cell will catch fire based on its neighborhood.

Spread probability distribution

To generate the burn history included in the synthetic data sets, a 5 parameter fixed balanced logistic function is used, See `Eq.3.5`. Feature vector [ω,γ,θ, ϕ] is given by $\vec{F}$.

$$\text{Logistic Model: } \text{logit}[p(\vec{F})] = \beta + 0.8\omega + 6\gamma + 0.2\theta - 0.2\phi \tag{3.5}$$

This probability distribution is sampled during fire simulations to build synthetic burn perimeters, as displayed in 3.3. Once this data was generated, our focus is to see how well the genetic program can evolve a set of functions that reproduces burn patterns. We compare three different models: a null constant model, a logistic model and an unrestricted algebraic model. The null model is composed of a single tunable bias parameter β_0 that is fit to the data. The model, thus, determines only the rate of fire spread, regardless of the neighborhood. Additionally, we consider a 5 parameter logistic model whose β parameters are evolutionary fit to synthetic data. The output of these functions are probabilities that a given site will catch fire.

$$\text{Null Model: } p(F) = \beta_0 \tag{3.6}$$

$$\text{Logistic Model: } \text{logit}[p(\vec{F})] = \beta_0 + \beta_1\omega + \beta_2\gamma + \beta_3\theta + \beta_4\phi \tag{3.7}$$

The unrestricted algebraic model is free to take any form given the bank of potential operators and terminals. The length of the expression is limited to a tree depth of 17 [58]. We will refer to this model as the genetic program model. Comparing these three models represents a good scope of expected performance. The constant model is a baseline, while the logistic model serves as an upper bound on the accuracy of the genetic program model, as it already has the same functional form as the underlying spreading function, and must only tune coefficients. All three models are evolved under the same set of hyper-parameters and learning schemes.

3.2.2 Implementation of Genetic Program

Validity and fitness of expressions are subsequently used to select ideal solutions and discard poor ones using tournament selection, with tournament size 4. The best performing individuals will be further subjected to cross-over and mutation. Evolution was implemented using the python library DEAP [59] according to the basic genetic programming as specified by [58].

An individual in the population represents a candidate probability distribution for fire ignition with fitness determined by how well it can reproduce a known fire event using the cellular automata.

Individuals are represented as syntax trees constrained to nodes of primitive oper-

ators and terminals. The operator set contains addition, subtraction, multiplication, protected division, negation, and basic trigonometric functions (sin, cosine). The terminal set is comprised of the features from of any given positions neighborhood (e.g. floating point values denoting that positions attributes: elevation, temperature, humidity, wind speed, and wind direction) as well as ephemeral constants in the range [-10,10]. Additionally, the tree is limited to a depth of 17. We impose this limitation to reduce code bloat and over-fitting, a common problem for genetic programs [60]. After a function is evaluated on a cells neighborhood features, a sample from the standardized normal distribution is drawn. If the output exceeds the sample, then the cell will ignite.

3.2.3 Evaluation of candidate models

Evaluation of the genetic program is conducted under two primary schemes: by evaluating over initial and final states of multiple landscapes, or by evaluating over each time-step of a simulation on a single landscape. This approach captures the ability of an individual to perform well at two timescales, reducing heterogeneity of solutions.

Experiment one: Learning from multiple landscapes

To calculate the fitness of an individual, the individual is used to simulate a set of fires across a set of landscapes. The burn simulation produces a predicted burn data set comprised of final states maps for each landscape. From the resulting data, the average intersection over the union (IoU) of the true and predicted state maps for each landscape is calculated. The IoU is commonly used as a cost function in reinforcement learning and image detection settings [61] [62]. The magnitude of the

IoU indicates how well the individual predicted the spread of fire in the allotted time window. To generate the reported experimental results, the GP was trained on 10 landscapes and tested on 10 additional landscapes. Training on multiple landscapes puts evolutionary pressure on solutions being able to generalize to different environments. This approach also prevents the GP from simply learning the Perlin noise distributions that generated the synthetic landscapes. We discuss and report the results of this experiment in `Figs. 3.4,3.5,3.6`.

Experiment two: Learning over single timesteps

Alternatively, we introduce another fitness function that considers how well an individual can train on one time-step to predict the next. One time step is defined as the period in which each cell on the landscape grid is considered for ignition once. In this way, we hope to capture (and subsequently evolve) the behavior of the wildfire *on that one specific landscape* at any given time-step rather than its behavior overall. For example, the evolving model is given the burn state of the first time-step of a ground truth burn for a specific landscape, asked to predict the second, then given a fitness equivalent to the IoU of that prediction with respect to the true burn state of the second time-step. These preliminary finesses are found using each time-step in the training data, then averaged to provide an overall measure. The only time-step omitted from this process is the last (as there is no subsequent time-step to provide a basis for calculating IoU). This approach was therefore attempted on a separate set of data than the first experiment, but this data was seeded, generated, and given a ground truth 'burn' using the same methods as the initial / final landscape state method.

3.3 Results

We first describe the behavior of fixed spreading distributions that are used to generate synthetic burn patterns. We then describe how well the genetic program, constant and logistic models performed under two experiments with the goal of reproducing the burn patterns.

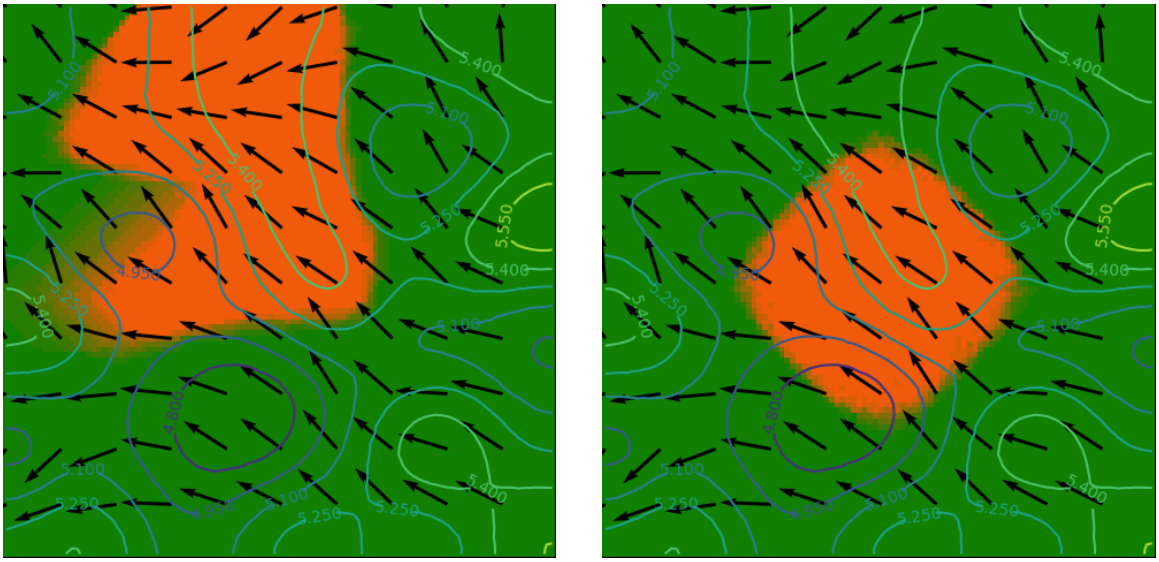
3.3.1 Behavior of biased spread functions

To design a function used to create realistic spreading behavior, we considered features one at a time, and visually analyzed their effect on spreading. Fig. 3.3 shows spreading according to three biased models and the result of the final balanced logits model as described in 3.5.

In Fig. 3.3c, we see the fire spreading uphill along a positive gradient of the landscape. Alternatively, in Fig. 3.3d, we see the fire following the wind current, moving North West. By balancing the contributions of the different features, we see the spreading behavior in Fig. 3.3a , with fire spreading in the direction of the wind current with discrimination to the elevation change. The balanced function accounts for all attributes of the landscape and generates more nuanced behavior.

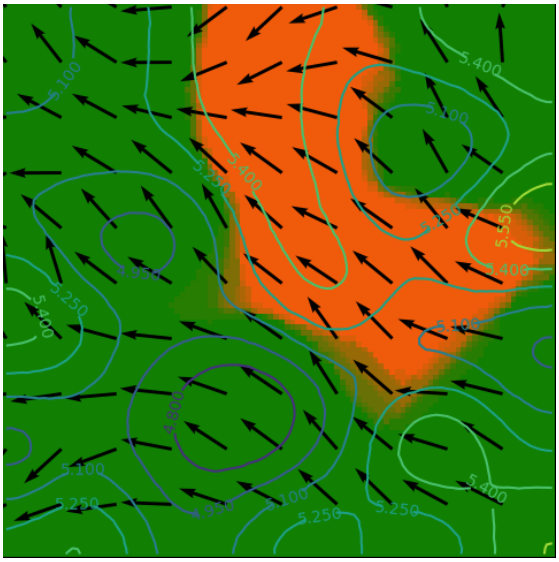
3.3.2 Model performance of training on multiple landscapes

Simulations were run with a training and test set both of size 10. The evolution lasted 50 generations with populations of size 100. This was done for both the constant and

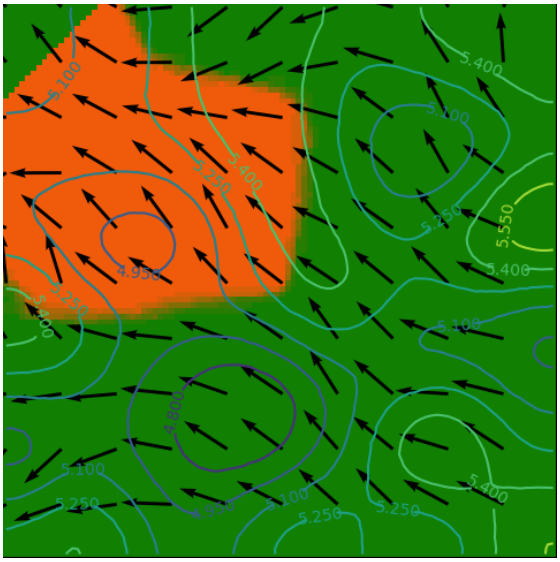


(a) Balanced function

(b) constant function



(c) Elevation-biased function



(d) Wind-biased function

Figure 3.3: Probability distributions that bias burn probability toward specific environmental features. The balanced distribution 3.3a gives equal weight wind and elevation. While the constant function 3.3b causes fire to spread stochastically in every direction.

logistic models, followed by the experimental GP model. Each used optimal hyper parameters found from parameter tuning as displayed in `Table 3.1`.

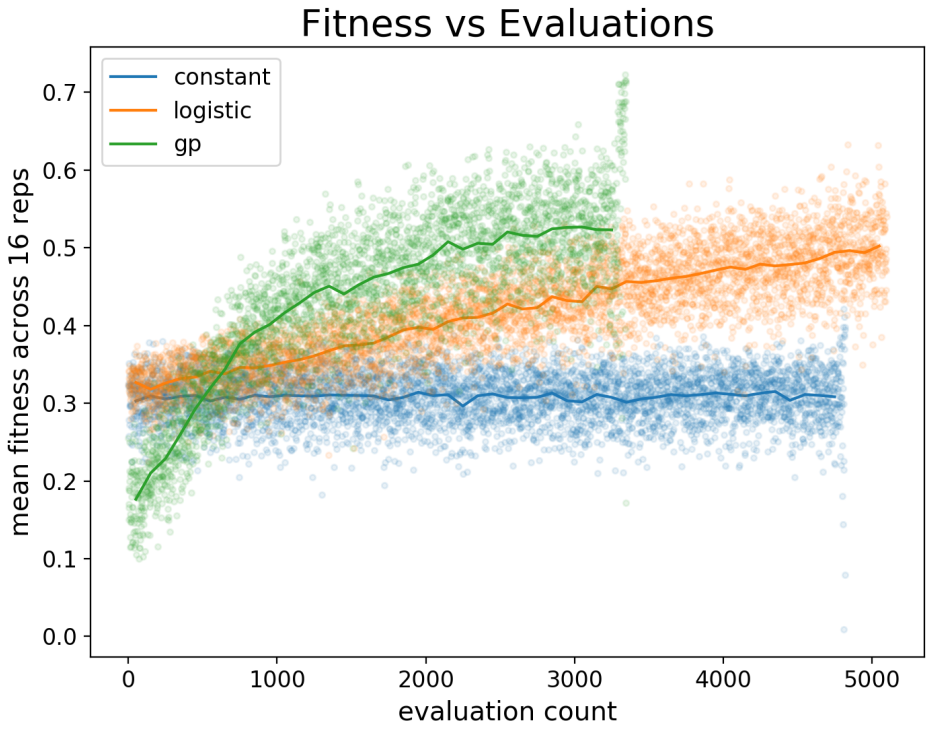


Figure 3.4: The GP model shows an initial fitness of 0.19, lower then both the constant and Logistic model. Both alternative models have implicitly bounded output while the GP must learn the correct domain over time.

In `Fig. 3.4` we see that the constant model shows no change in the distribution of mean fitness of new individuals. Alternatively, the logistic model shows an upward trend; individuals resulting from crossover or mutation thus improved in mean fitness. The genetic programming model demonstrates poor fitness in early function evaluations but very fast improvement.

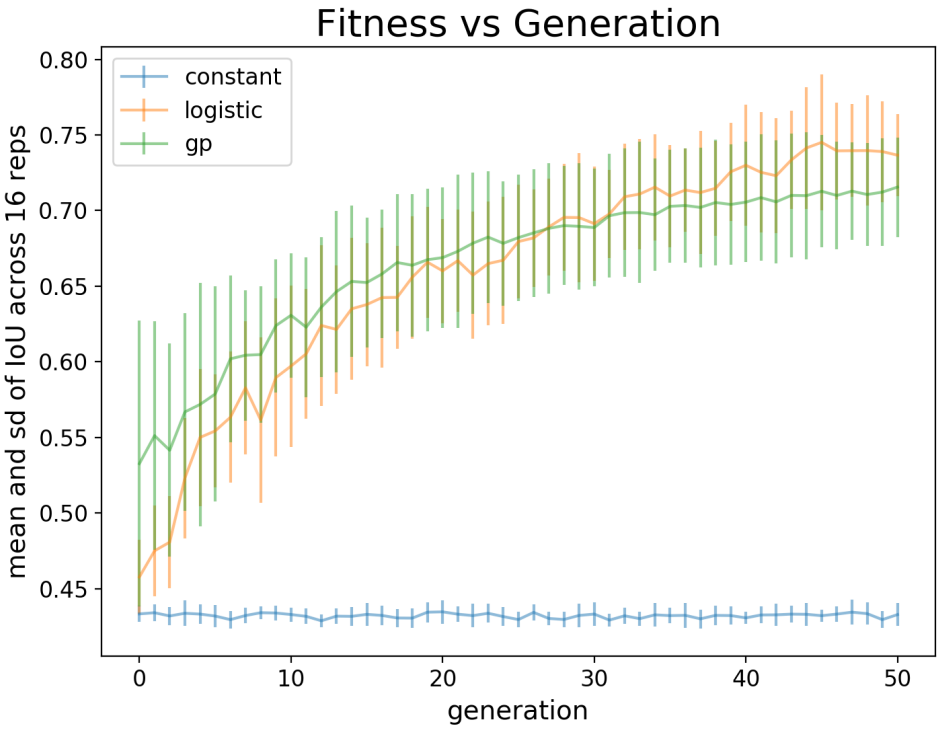


Figure 3.5: While the GP model initially displays a faster learning rate, the two models converge in performance after 26 generations. The constant model indicates a reference baseline performance of 43%, constant over generations.

Fig. 3.5 shows the mean and standard deviation of the best fitness of each generation, across 16 repetitions.

We expect in the long run for the logistic model to eventually find the right logits to match the true spreading model. Once this happens, the GP will have a difficulty competing because its solutions are much more complex. We discuss methods to reduce this complexity in the discussion section.

`Fig.3.6` presents the distribution of maximum fitnesses per repetition from the three models in addition to the ground truth model. The true model is the "balanced logits" model that was used to generate the burn. Due to the stochastic nature of

the burn simulation, this model fitness represents optimal fitness. The logistic model comes close to the performance of the true model, as expected. The constant model represents the performance of an extremely simple model. We have separated the training and testing results. While typically, the training results in evolutionary algorithm methods are worse than the validation results, we believe that the variance in the environmental layers me be favoring the validation set.

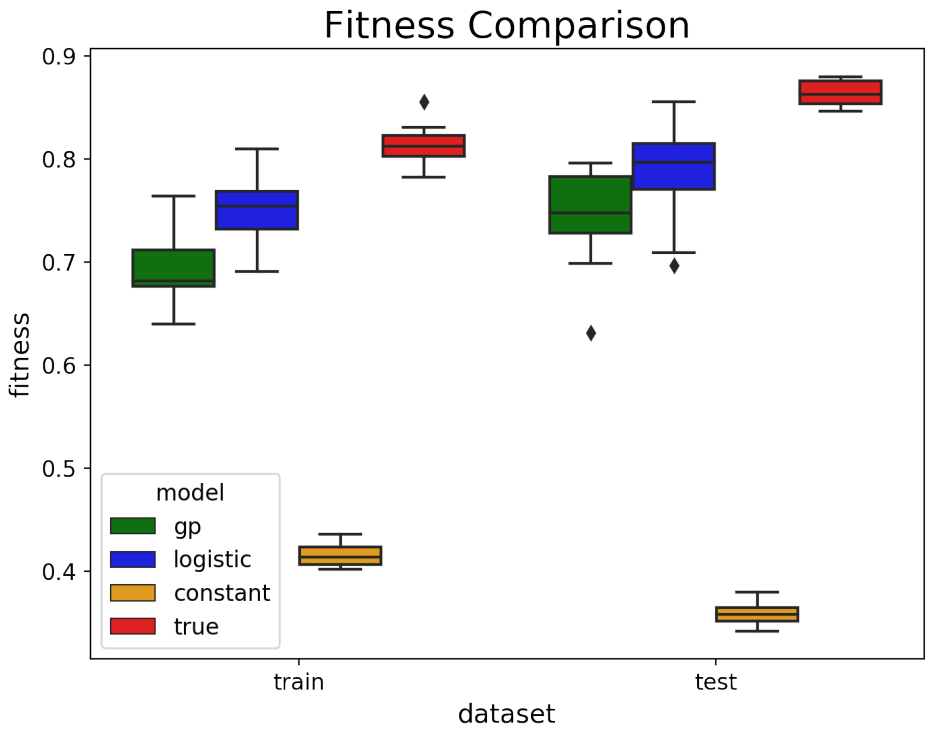


Figure 3.6: The Logistic and GP models come within %10 of producing the same burn pattern as the underlying spreading function. This indicates that the GP has learned the short term spreading pattern in this environment.

3.3.3 Single landscape, multiple time-step evaluations

Another characteristic of a well fitting spreading distribution is the ability to foresee short term changes in the fire front as there are a number of ways that a fire could burn to the final perimeter. We employee the same evolution scheme as the prior experiment but use a different cost function to drive evolution. Individuals are evaluated for fitness after any single time-step of the simulation. We run a simulation according to an individual distribution an evaluate its success at predicting one time step ahead by taking the IoU between the predicted burn set and the true burn set.

We use a population of 100 individuals, each run for 100 generations within each repetition. Again, the results of each repetition were saved, yielding a total of 20 runs from which data could be extracted. Mutation and crossover rates were 0.08 and 0.8 respectively. We first examine the fitness with respect to each generation, see Fig. 3.7:

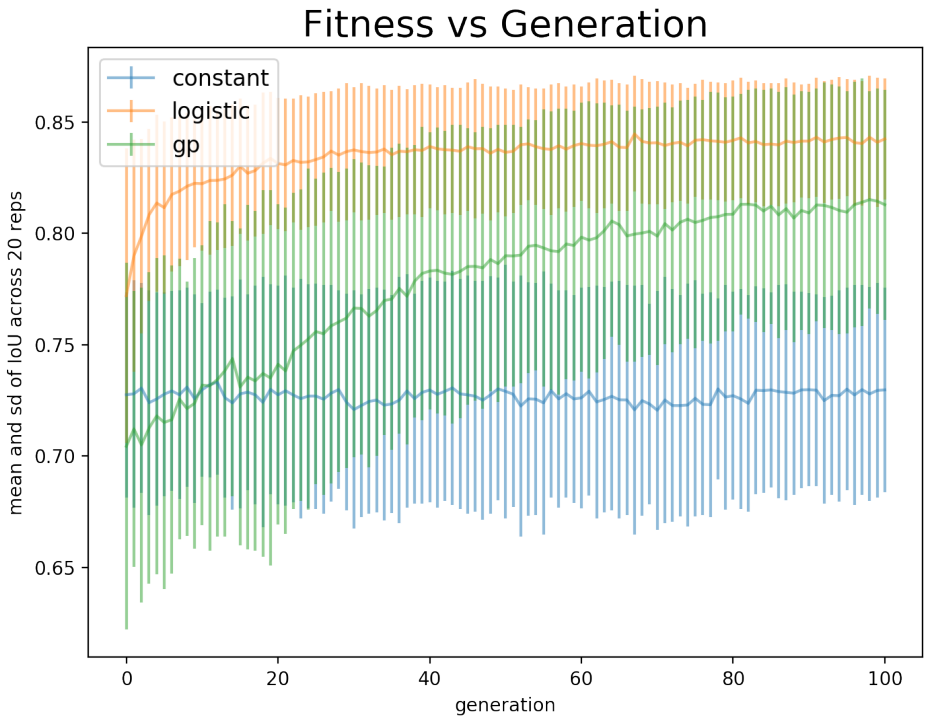


Figure 3.7: The logistic model learned to 83% accuracy within the first 20 generations. The GP shows a much slower learning rate but approaches the same average fitness. All models display highly variant results as indicated by present error bars.

The constant model did not change fitness throughout the entirety of the evolutionary run and with each passing evaluation or generation. However, the GP and logistic models demonstrated different behavior than reported in the first experiment, with the logistic method outperforming the GP model in both number of evaluations to convergence and overall maximum fitness reached. Even considering the variance of each model's data, the logistic method demonstrated a far stronger advantage in the multi time-step experiment.

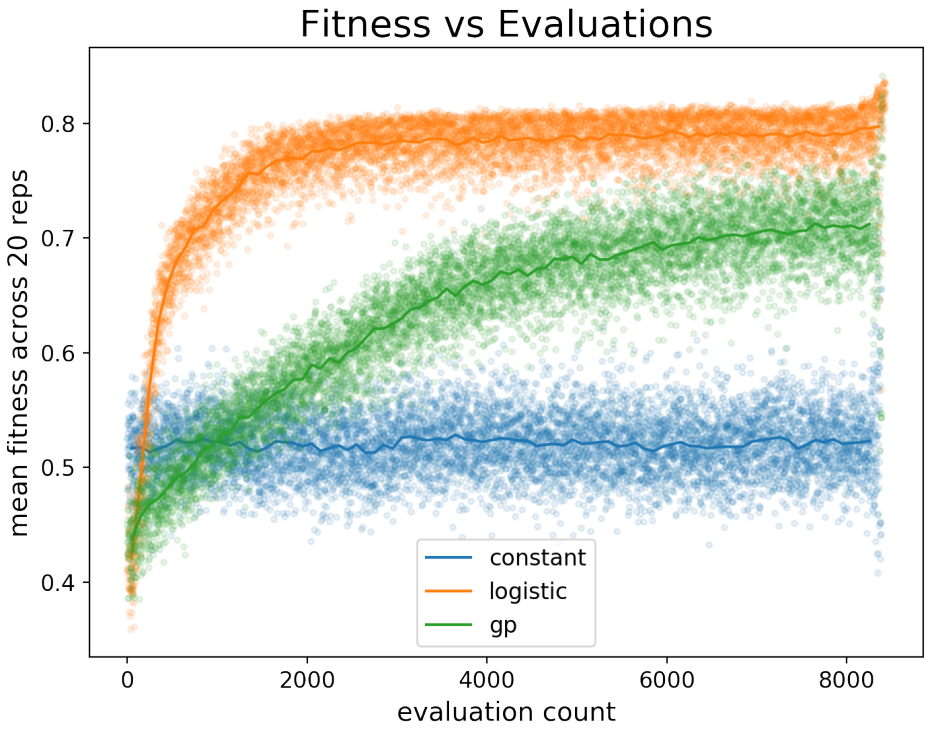


Figure 3.8: The constant model demonstrated abnormally high performance; even outperforming the GP model in the beginning of the evolutionary run. The GP models grows in fitness over evaluations and approaches the performance of the logistic model.

The constant model demonstrated expected performance, while the logistic and GP models performed similarly to one another. However, the fitness over generations showed high variance in in `Fig. 3.7` with error bars covering just over a full tenth of the fitness scale (0.1). Furthermore, we examined the overall performance of the models over the course of 20 repetitions in Fig. 3.9.

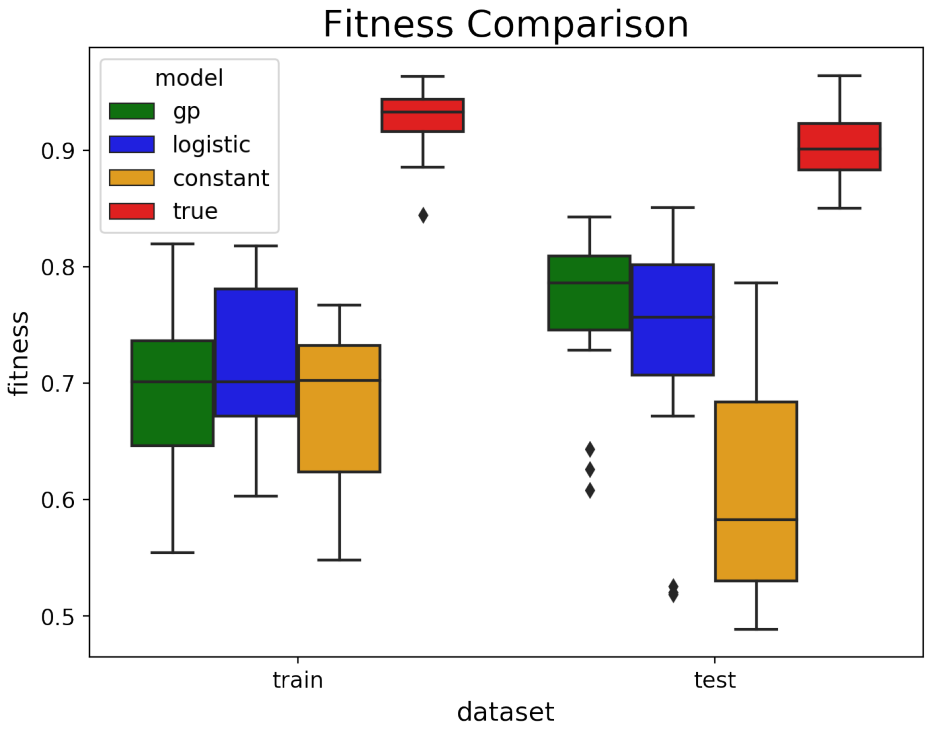


Figure 3.9: The constant and GP models demonstrated comparable training fitnesses, while the logistic model dominates. The true distribution is used to recreate a 20 fires to represent the true stochastic upper bound on performance.

While being tested on unseen environments, the constant model performed significantly worse, while the GP and logistic model performed comparably.

3.4 Discussion

We developed a model that learns the spreading behaviors of synthetic wildfires based on environmental, atmospheric data coupled with historic fire burn perimeters. These data-sets can be synthetic or real. We have shown that the macro spreading behaviors can be learned by evolving the spreading function at differing temporal resolutions. We show that the uninitialized population of algebraic expressions can evolve to

produce prediction accuracy's comparable to the true underlying spreading function. We will next discuss some of the structural components of the evolution process.

3.4.1 Parameter tuning

An essential part of optimizing evolutionary algorithms is setting the correct hyper parameters. We choose to consider mutation and crossover for tuning. Using a grid search, crossover and mutation rate were both tuned to optimize final fitness on a held out validation set. A 5 fold cross validation was used. These parameters were tuned for the constant and logistic null models under an initial/final landscape fitness function scheme.

The genetic program was tuned by finding the optimal crossover rate sweeping over values $[0.5, 0.6, 0.7, 0.8]$ while holding the mutation rate constant at 0.1. The optimal crossover rate was then used to find the optimal mutation rate sweeping over values $[0.1, 0.2, 0.3, 0.4, 0.5]$ with the same experimental design described in Section 2.3. In future work these hyper-parameters would also be tuned with a grid search.

Table 3.1: Optimal Hyper-Parameters

Model	Crossover Rate	Mutation Rate
Constant	0.7	0.8
Logistic	0.4	1.0
GP	0.8	0.08

We note that the optimal mutation rate for the Logistic Model is 1. This indicated that the Logistic model is primarily evolving from selection and mutation.

Additionally in the presented result sets, the constant and logistic model often held initial fitnesses significantly higher then the the GP model. We determined

that this was due to the GP needing to learn the optimal distributions of outputs to become a true probability distribution. Initial distributions can feasibly contain negative numbers resulting in no spreading. Alternatively, the logistic model will implicitly produce a distribution bounded in $[0, 1]$.

3.4.2 Optimal Function Forms

While the accuracy distributions of the fittest indicated that the GP can learn a function that will reproduce the spreading patterns, we are also interested in the functional form of the solutions and how close they are to the balanced logistic function. We track the mean and max length of the fittest individuals over 20 reps for 100 generations. The results are displayed in Fig. 3.10

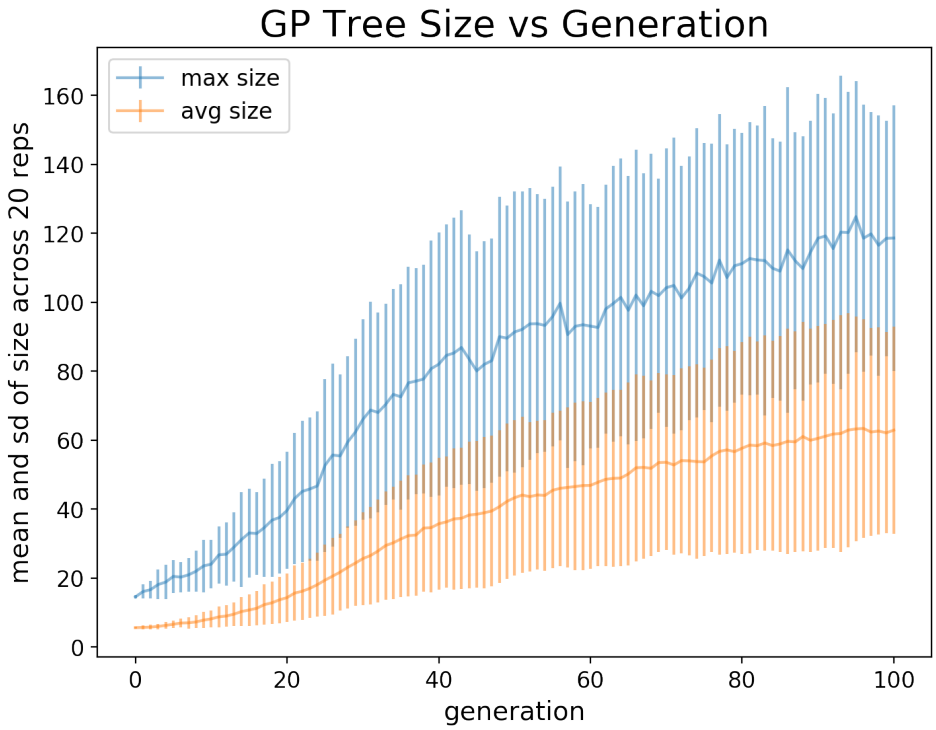


Figure 3.10: Over time, the size distribution of solution increases. This type of trend can indicate code bloat. However, there is a reduction in size acceleration after the 40 generations.

We note that as the population evolves, the individuals grow larger in length. Code-bloat is a problem common to genetic programs and can lead to over fitting and loss of model generalization. To reduce this problem, in future iterations of the project stricter tree depth or node count limitations could be enforced. A fundamental problem present at this resolution of simulation is heterogeneity of solutions. I.e. multiple solutions can potentially generate the same behavior, achieving the same fitness. Heterogeneity makes uncovering any causal relationships very difficult. We present two example expressions sampled from the final evolved population.

$$\text{True Solution} : \text{logit}[p(\vec{F})] = -7 + 6\omega + 0.2\gamma + 0.2\theta + 0.8\phi$$

$$\text{Fitness} = 0.80 : \text{logit}[p(\vec{F})] = -3.88 + 0.43\omega + 5.25\gamma + 0.10\theta - 0.43\phi$$

$$\text{Fitness} = 0.85 : \text{logit}[p(\vec{F})] = -7.52 + 7.15\omega + 0.22\gamma + 0.27\theta + 0.87\phi$$

We see that both evolved solutions produce high fitnesses but use different coefficients. While this may be suitable for pure prediction tasks, we note that this is a drawback to this method. We conjecture that this may have to do with the simplistic nature of the synthetic data-sets. Further, the the full genetic program produces wild results as shown below.

$$\text{True Solution} : \text{logit}[p(\vec{F})] = -7 + 6\omega + 0.2\gamma + 0.2\theta + 0.8\phi$$

$$\text{Fitness} = 0.71 : (-0.32 + \phi) * \phi$$

$$\text{Fitness} = 0.79 : \frac{\omega^4 * \phi}{\gamma} - 9.76 * \theta$$

These solutions do not closely resemble the true solution, despite having high fitness values. In future iterations, we would like to constrain the complexity of the solution as a second objective to fitness.

3.4.3 Limitations and Future Steps

As is the case with many models, there are many assumptions that are held in this model that could be relaxed with additional environmental layers. Most importantly, we have considered fuel sources to be homogeneous and all landscapes are comprised

of tree fuel beds. Of course in reality, there are complex distributions of fuel types and this can have a huge effect on fire spreading behavior. Additionally, we assume uniform tree height, which has also been shown to be an important factor in heat transfer and material ignition [63]. We also assume that fire can only spread between neighboring cells on the grid; however, embers can spread to disconnected patches of vegetation starting "spot fires". While these assumptions are clear limitations to the applicability of the model, adding these features to a future model is highly feasible and would not introduce a noticeable increase of complexity.

The obvious next step of this project is further optimization, then validation of the method using real wildfire data. There are several suitable datasets that are available to validate this method, including the 2011 Richardson Wildfire and 2016 Fort McMurray wildfire, both of which took place in Northern Alberta. These data sets are openly available through the NASA's EarthExplorer Data Portal [64]. The accompanying weather data is available through the Canadian Weather service [65]. The results of this experiment could then be directly compared to the recent work by [34]. These two fires serve as a perfect train and test set as they took place in a very similar climate at different times.

Additionally, we hope to apply both experiment one and experiment two as a joint multi object fitness function. This way, individuals that can do short and long term prediction are selected for evolution.

3.5 Conclusions

We propose a genetic program embedded inside a cellular autamata simulating wildfires in different synthetic landscapes. We found that the genetic program is able to capture the behavior of the wildfire to produce burns on synthetic data sets that are realistic to burns generated by the underlying spreading function. We summarize some of the main takeaways from this work.

- On average, the GP is well suited to recapture the spreading patterns produced by the balanced logistic function. The GP produces average accuracy's within 15% - 30% of the true spreading function for experiments 1 and 2 respectively.
- Macro spreading behaviors can be learned by tuning the spreading function at differing temporal resolutions
- Evolved solutions are subject to code bloat and do not represent the realistic driving rule-set.

While some of the typical problems with black-box prediction are still present in this model, it is exciting to see that synthetic spreading behavior can be predicted with a moderate accuracy.

This research adds to a new avenue for evolutionary methods to learn spreading rules for cellular automaton simulating spreading events based on geo-spatial datasets. In the future, we would like to validate this method on a data-set of ground truth remote sensing atmospheric and historic fire perimeter images. Further

The source code for this project is available at the public github repository at : https://github.com/maxfieldEland/EC_2019_FinalProject

Chapter 4

Case Study 2 : Deep Learning and 24 Hour Front Prediction

Branching off of the work recently done by Radke et al [33] in developing the DeepFire model, we propose a data driven fire simulation based on a deep convolutional neural network that predicts the evolution of a fire front in 24 hour periods. We attempt to reproduce the results reported from the FireCast model publication and generate notable improvements using an alternative model architecture and prepossessing step that we believe will help the model learn the fire behaviors. Additionally, we produce an atmospheric geo-spatial API that can automatically collect, curate and pre-process all input data from their native third party web databases. This research offers two primary products, a statistical model for forecasting wildfire spread and a fully functional API with an example curated data set. The RESTful API can be used in adjacent research efforts to produce clean geo-synchronized digital elevation models, satellite spectral bands, land cover maps, atmospheric history and time series of wildfire perimeter maps.

As this type of modeling is quite data intensive, an API that pulls together disjoint data sets that may use differing projection systems,encodings and meta data is a great resource to not only our team but also future analysts and modellers. The API documentation is included in the open source repository for this modeling project.

4.1 Deep Learning and Fire Perimeter Prediction

Deep learning models that leverage the abundance of remote sensing data available have made a recent impression on wildfire modeling [33], [34]. Modern models are becoming capable of maintaining the precision and accuracy of traditional physical methods while offering more flexibility towards learning different environmental regions and timescales, solving two of the problems that historic fire models have faced. Models discussed in the literature review largely suffer from inflexibility and require a large overhead to be tuned to multiple climates.

Further, applications of deep learning in atmospheric weather events such as precipitation [66–68] have seen recent success. The work of [69] is particularly relevant to modeling wildfire spread over time. The MetNet algorithm outperformed current state of the art fully physical models using only historic local atmospheric conditions and topography. Employing recursive layers to capture the temporal features of precipitation events and axial attention to encourage the model to focus on pertinent bounding boxes within the image based feature set, the model was able to learn high dimensional and robust features. MetNet serves as an example of the power of deep neural models to make generalized long term associations between remote sensing

data and natural physical processes.

4.1.1 A Review of FireCast

The FireCast system is a data driven wildfire spreading model that learns spreading behavior by implicitly associating the wildfire perimeter changes with atmospheric conditions, spectral images and elevations. The model learns to predict the state of a geolocated pixel from a satellite image after 24 hours based on a $300m^2$ sized neighborhood of states and environmental characteristics. Atmospheric conditions include temperature, wind speed, wind direction, humidity, dew point, atmospheric pressure and precipitation. These data are sourced from the National Oceanic and Atmospheric Association (NOAA) [38]. In addition to atmospheric conditions, the model uses a digital elevation model to account for spatial variance in elevation. The digital elevation model is used to derive a landscape aspect in which each location contains a degree from north for the direction the ground is facing. Additionally, four of the eight spectral layers captured by the Landsat 8+ satellite instrument are used. The four layers include the red, blue, green and infrared spectra. The combination of these satellite bands have been reported to represent abstract vegetation health indexes, such as he Normalized Difference Vegetation Index (NDVI) [70]. The ground truth fire perimeters are geo-located shape files. For a given fire in the training set, the model is shown the initial fire perimeter and the fire perimeter after 24 hours. A subset of examples are withheld for validation purposes. FireCast is used to make predictions about fires in a small region of the Colorado Rocky Mountains. We hope to extend this model to several climate types within the Sierra Nevada's of California. This area encounters a growing severe fire season each year. We aim to

reproduce the dataset nature used to train the FireCast model as close as possible to maintain comparability in results, however there are a number of data preprocessing steps that are not discussed in the publication. FireCast produces remarkable results in comparison to FARSITE and other industry level simulation software. FireCast produced significantly high F-Scores than Farsite while predicting a 5 day wildfire in the Rocky Mountains of Colorado.

4.2 Data Curation

We have curated a dataset that covers a large span of wildfires in the Eastern Sierra of California, an area that is heavily effected every year by disastrous wildfire events. The data set in its final form is a cleaned, processed and geo-synchronized collection of LiDAR(Light Detection and Ranging) , satellite spectral layers, fire perimeter maps and timely atmospheric conditions. This data set can be used as a training and testing set for learning models in addition to an analysis of correlation between recorded perimeter dynamics and environmental conditions. The overhead for collection of these data is comparatively extensive and its open source availability provides future modellers to train and validate different forecasting systems. The LiDAR data used in this study is aggregated into a digital elevation model with a 30x30m spatial resolution. The raw elevations are provided by The National Map [64] which produces a large variety of geo-spatial data products.All LiDAR products posses a 1/3 arc second resolution in the form of a GeoTiff files. Each pixel in the data product is referenced in UTC (Universal Trans Mercator) coordinates. During preprocessing, this coordinate systems is reprojected into the local ESPG latitude and longitude

projections of the accompanying area. The spectral layers aggregated for this project are originally captured by the Landsat 8+ satellite instrument that orbits the globe on a 99 minute period. However, the instrument passes over our area of interest about every 16 days. While the geospatial layers may not be stored in the same data structure or represented in the geographic projection, they are simply matrices storing geo-located floating points.

In some cases, the fire perimeter may spill over between satellite image tiles, in which case up to four tiles are aligned and joined to build a mosaic raster. The same is true of the DEM.

After collection, for a given fire event, the data layers are stacked atop one another using a common EPGS projection. We crop the data stack to the extent of the largest stage of the fire perimete including a 300m buffer on all sides. Between different fire data-sets the final dimensions of the data stack will vary depending on the maximal extend of the fire.

Data collection is a common difficulty in data intensive deep learning applications. The DeepFire API will lift this weight for future research efforts in this area. The API allows a user to specify the approximate latitude and longitude of the historic fire and a time series of fire perimeters, atmospheric conditions, satellite spectral images and a digital elevation model are returned.

4.3 Methods

We describe the generation of training data using a weighted spatial sampling method as well as the architecture of the neural network being trained.

4.3.1 Data Preprocessing

Prior to model input, all input features go through a normalization process to optimize model performance. In this case, data normalization is used to scale the data into the same range of values for each input distribution.

All input features are normalized by the transformation specified in Eq. 4.1, which is commonly referred to as the Z Normalization [71].

$$\frac{(x - \mu_x)}{\sigma_x} \tag{4.1}$$

By using this particular transformation, we are affording a light assumption that these time-series are in fact stationary. However, it is more likely that they are in a wide sense stationary, there are physical upper bounds to these processes, although they may not be observed in the training set.

An exception to this treatment is the wind direction variable. As opposed to calculating the Z feature, we convert the source degrees to radians and normalize with the $cos(x)$ transformation as indicated in Eq. 4.2.

$$X' = \cos\left(\frac{x * \pi}{180}\right) \tag{4.2}$$

4.3.2 Stochastic Weighted Spatial Sampling

Post prepossessing, the input data is drawn from the data set by performing a weighted random sample from the landscape. At least 60% of the total image is sampled for each fire data set. A spatial weight is imposed on the point sampling that

encourages points near the perimeter of the fire to be sampled over points far away from the perimeter. The weighting distribution is drawn from a smooth Gaussian and all points are still possible to draw. Sampling is performed without replacement, this way, it is impossible for the same location to be drawn twice from the same time step.

We will refer to these sampled points as "points of interest" (POI). Once POI has been selected, its 30 pixel neighborhood is stored, as displayed visually in Fig. 4.1. For training and validation, a set of neighborhoods and their corresponding POI label in the next time step is sampled from every fire in the data set and from every available 24 hour period.

To generate additional data and variance in the data set, we perform data augmentation on the data stacks. Transformations are currently limited to rotation and mirroring.

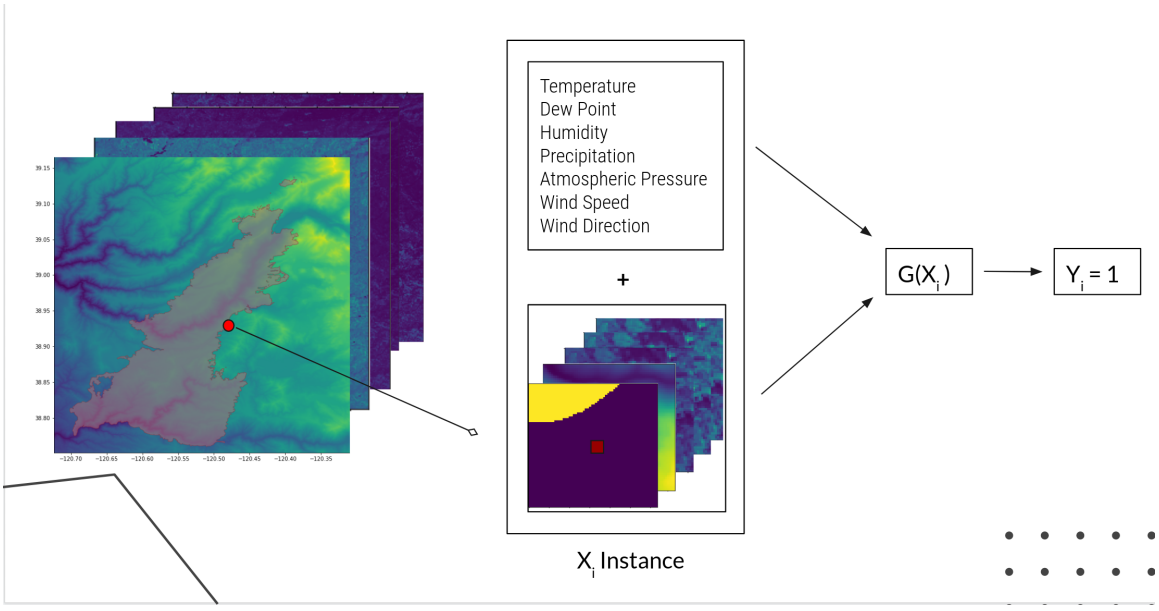


Figure 4.1: Example sample instance from the King Fire near Lake Tahoe CA.

These neighborhoods are the input to a deep convolutional neural network that is learning to predict state of the center pixel 24 hours in the future.

Model Architecture

The model architecture was initially inspired by the model reported in the FireCast publication [33]. The FireCast architecture is a series of 3 convolutional layers that are eventually flattened and then concatenated with a 1D array containing the atmospheric information. A series of dense layers are finally passed through a sigmoid function which provides the prediction value for the sampled pixel of interest. The FireCast publication reports very high validation accuracy's, up to 95% when predicting the perimeter dynamics of unseen fires during short prediction windows.

However, we found that the performance of the particular architecture was not very high and under-fit the data. This is likely not due to misreporting in the publication and likely due to differences in data preprocessing and unreported model hyperparameters. Moving from the FireCast architecture, we proceeded to add parameters until the model was over-fit. By adding regularization,data augmentation and dropout, the performance of the model improved greatly. The model used to generate results in Figs 4.2a,4.2b,4.4a, 4.4b is described by Table 4.1.

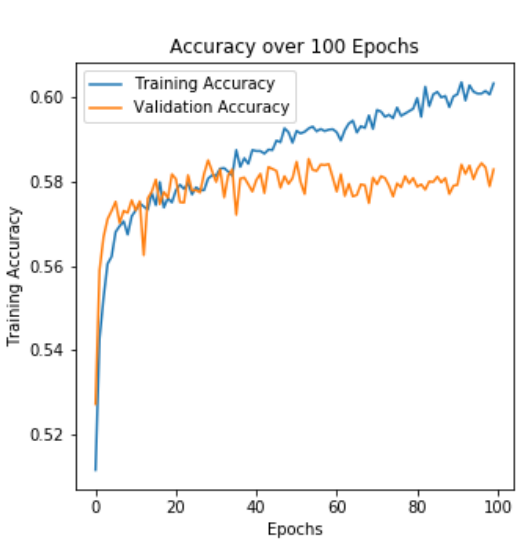
Layer	Operation	Kernel/Pool Size	Feature Maps
1	Convolution	3 x 3	32
2	Convolution	3 x 3	32
	Max Pooling	2 x 2	-
3	Convolution	3 x 3	64
4	Convolution	3 x 3	64
	Max Pooling	2 x 2	-
	Dropout	-	-
4	Convolution	3 x 3	128
5	Convolution	3 x 3	128
	Max Pooling	2x2	-
	Dropout	-	-
6	Convolution	3 x 3	256
7	Convolution	3 x 3	256
	Flatten	-	1152
	Concatenate*	-	-
8	Dense	-	264
9	Dense	-	128
10-out	Dense		1

*Table 4.1: The DeepFire Model Architecture is based on FireCast with a few alterations adding non-linearity and using a max pooling downsampler instead of an average pooling.*Atmospheric tensor is concatenated directly to flattened output of the 7th convolution layer*

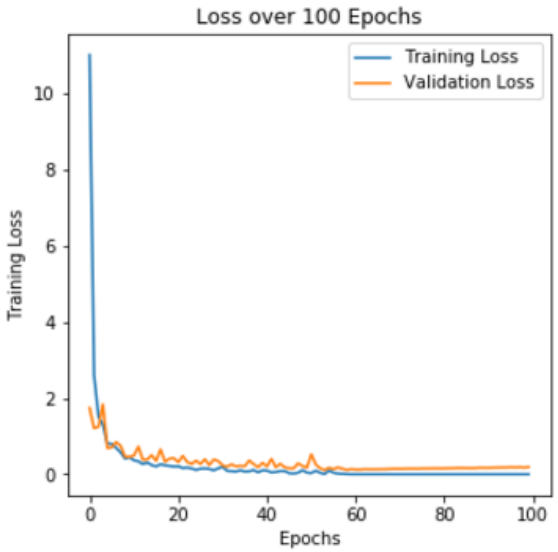
This model was implemented in Python 3, using the Keras and Tensorflow Deep Learning Frameworks [72].

4.4 Results

Initial reported results are generated by training the model on 24 hour intervals sampled from all time steps of the recorded King Fire of 2014 in the city of Pollock Pines, California. Figs. 4.2a, 4.2b show the accuracy and loss as the model learned on batches of training data. As a first pass, this result indicates that the model is able to learn. Moreover, this task is quite difficult as there are potential spreading properties that are expressed in the early, middle and late stages of the fire uniquely.



(a) Training accuracy shows a promising upward trend while validation accuracy shows an early convergence at approximately 0.58 accuracy. This suggests an over-fit model.



(b) Both training and validation loss drop towards zero. We see that the validation loss is unstable throughout

While this initial result is not very promising, it does provide a proof of concept and a launch point to diagnose some of the problems that the network may be experiencing in learning semantic associations between the input and the fire perimeter evolution. The over-fitting of the model suggests that there are enough parameters

present. The confusion matrix describing the models predictions on the validation set in provided in Fig. 4.3.

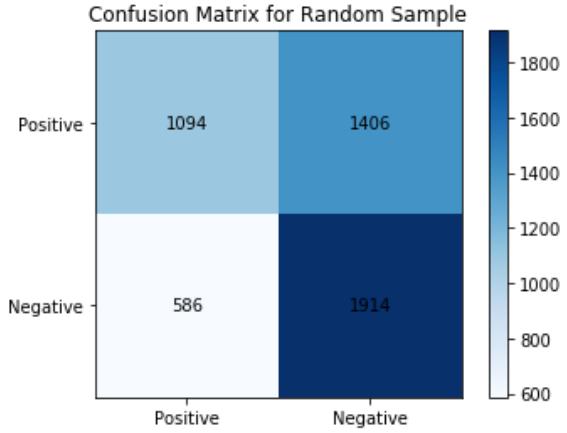
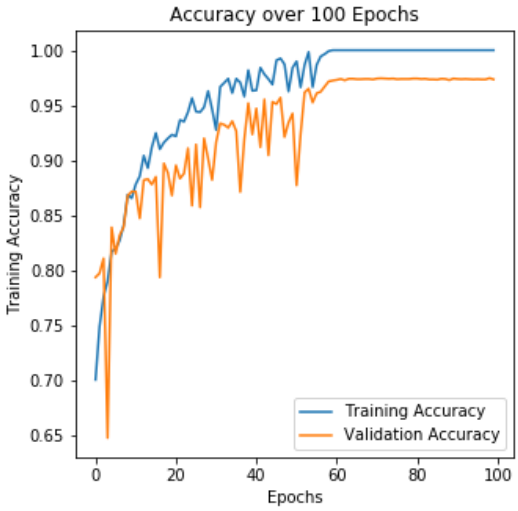


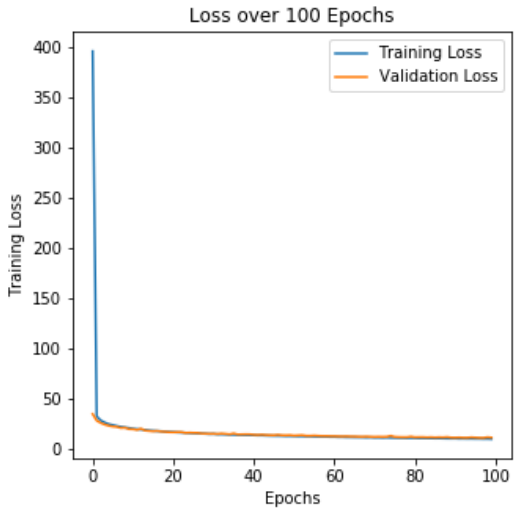
Figure 4.3: Confusion matrix for model trained across fire duration. Model shows very poor performance and appears to favor false negatives.

We see that the model favours false positives. We note that this may be explained by the difficult class balance as discussed in Fig. 4.7. The next step is to re-frame the question in a way that is more tractable to gradient descent. To make the problem a little easier, we split training sets by the day and ask the model to learn the dynamics unique to only one time period. The learning curves for a model trained on a single random 24 hour periods are presented in Fig. 4.4a and 4.4b. We hypothesize that there are dynamics unique to the changes of the perimeter at different stages of the fire. Thus, the loss landscape is much easier for the model to navigate through gradient descent for a single time period than it is for all days of the King Fire.

These results indicate that the model is capable of learning short term associations. Both the validation and training learning rates fully converged. The next step is to examine how the model trained an a given time period will generalize to an unseen period. A confusion matrix of the models predictions on the validation set is presented



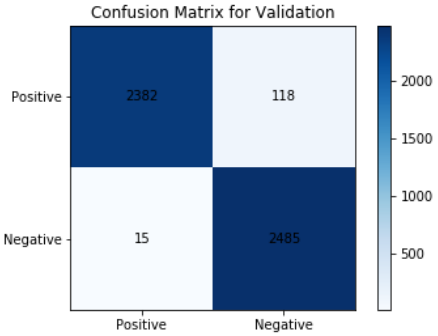
(a) Training accuracy fully converges to an accuracy of 100% while validation converges slightly below at 97%.



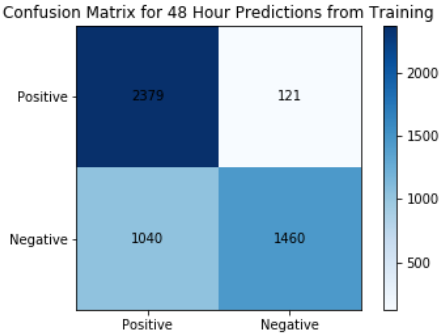
(b) Both training and validation loss drop to zero instantly

in Fig. 4.5a. Further, we use the model to predict the 24 hour evolution from one day ahead of the training period. The results of this experiment are present in Fig. 4.5b.

The results in Fig. 4.5aare from predicting the fourth day of the King Fire from the conditions on the third day. The model shows relatively few incorrect classifications, with 118 false negatives and 15 false positives. Alternatively, the results in Fig. 4.5b demonstrate the ability of the model to predict the fifth day of the King Fire, from the conditions of the fourth with the weights learned from the dynamics of the 3rd-4th 24 hour period. The model performs worse at this task, with severe false positive predictions. However, this result is particularly insightful when we look at some of the differences in the change in fire perimeter area between the three days considered. This discussion is further examined in the Discussion section in conjunction with Fig. 4.7.



(a) Confusion matrix from validation set prediction results.

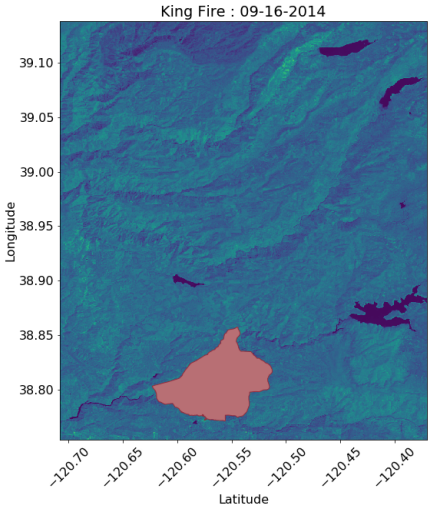


(b) Confusion matrix for predicting fire perimeter on 09/15 based on input from 9/14 from model trained on only 9/13-9/14 data.

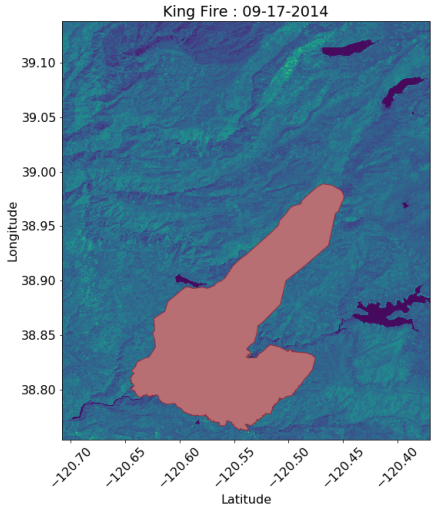
4.5 Discussion

4.5.1 Class imbalance and training bias

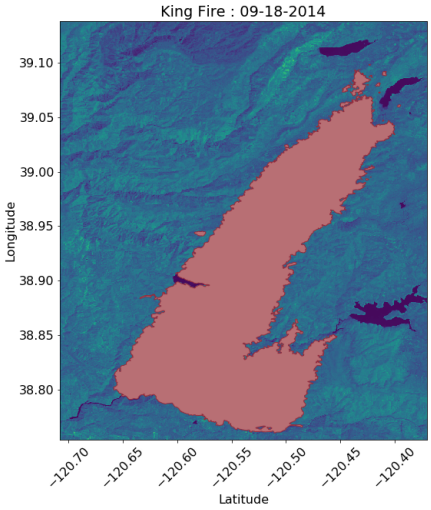
One ongoing question about the sampling strategy is how to best consider the class imbalance. The FireCast publication reports that a fully random sample was taken from the landscape. The true class balance from each training fire is unique and dependent on how much of the landscape the full extent of the fire occupies. The training space is defined by a bounding box that buffers the largest extent of the final fire perimeter by 30 pixels. Thus in many cases, a fire will occupy a small portion of the sample space for the beginning of the data time series.



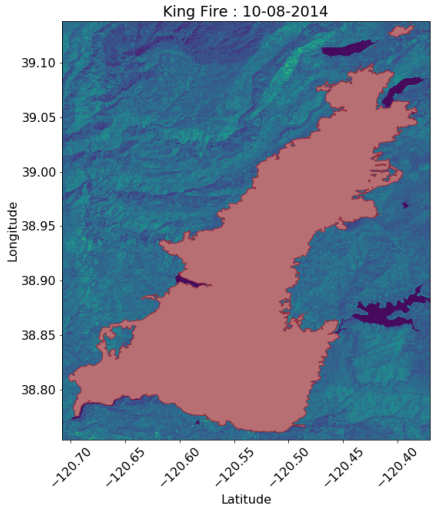
(a) September 16



(b) September 17



(c) September 18



(d) October 8

Figure 4.6: Example crop for the California King Fire of 2014. The final extent of the fire shown in 4.6d. This perimeter map represents $97,000$ *acres burned as a result of the fire event. Figures 4.6a,4.6b, 4.6c show the development of the fire over the 2nd, 3rd and 4th days.*

The sample space from which training data is drawn is somewhat arbitrarily constricted. The size of this space around a wildfire directly effects the corresponding

class imbalance. Four example time steps of the 2014 California King Fire are presented in Figure 4.6. As the images are cropped to the extent of the largest fire, the class balance shifts over time. Note the difference in change between September 16 in 4.6a and September 17 in 4.6b and the change between September 17 and September 18 in 4.6c. In a single day, the class balance changes by over 20%. Alternatively, previous and subsequent days do not contain such a high change in burn area.

In the case of the King Fire, the sample space at a maximum would be is restricted to a 2000x2000 cell grid. The initial fire occupies very little of the sample space while the final fire occupies approximately 30% of the sample space. Figure 4.7 show how the area of one fire pixels changed over the extent of the King Fire event.

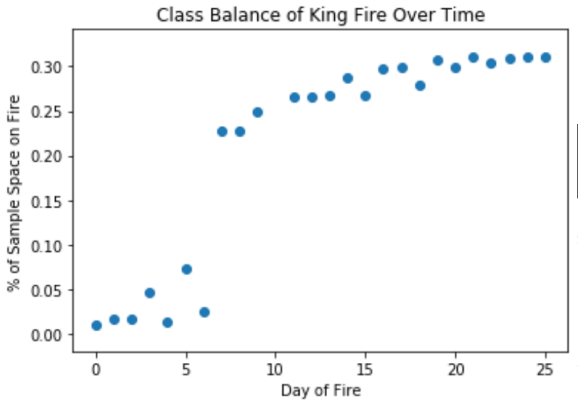


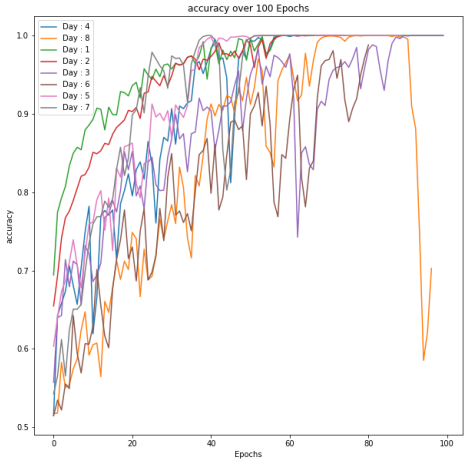
Figure 4.7: Changing class balance within the sample space of pixel values representing the King Fire.

We see that as the fire burns a majority of its total area within the first 7 days, leaping from 8% burn area to 23% burn area in one day. We further examine the difficulty of predicting the fire behavior during that day of fast spreading. See Fig. 4.9d for the learning curves of a model trained on just the We also note the reduction in area. This aspect of the data is not currently understood as the interior of the fire never decays. However there are instances in which the perimeter fluxes, creating

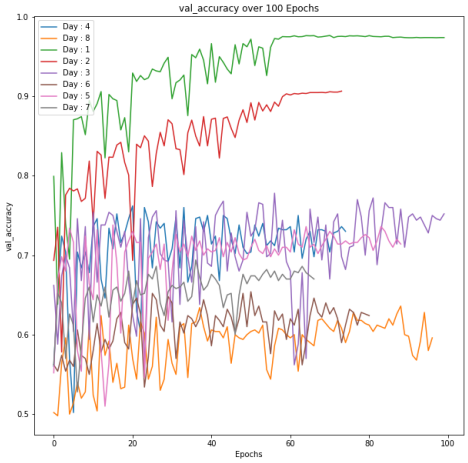
more or less area within the shape. We have contacted the folks that prepared the GeoMAC [73] data to better understand the process of shape creation and await a response.

Diagnosing the Network

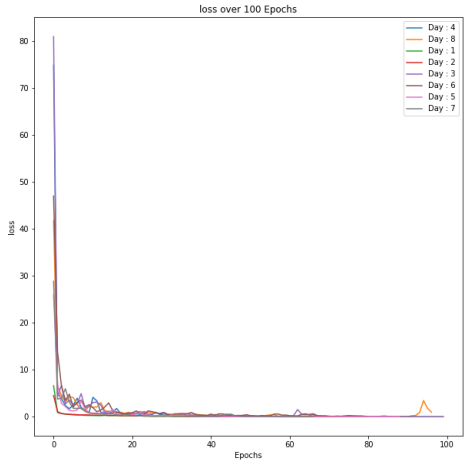
To determine which components of the problem are readily tractable to gradient descent, we break the general problem into smaller sub problems by eliminating the temporal generalization requirement. By training eight models across eight twenty four hour periods and comparing their validation performance, we find that the large changes in fire diameter in a given period are correlated with worse results. We visualize these results in Fig. 4.8.



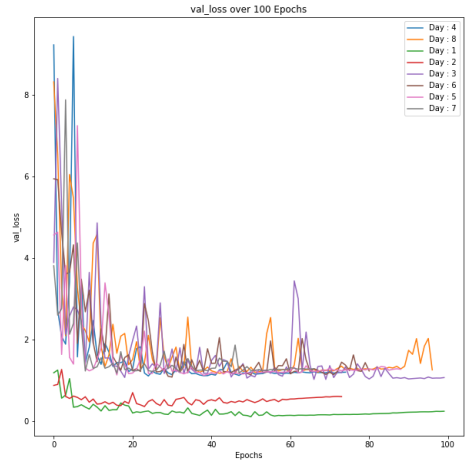
(a) Training Accuracy



(b) Validation Accuracy



(c) Training Loss



(d) Validation Loss

Figure 4.8: We see that the model struggles with the samples that contain greater differences. There is also high instability present in both the training and validation accuracies, which is reflected in the loss minimization.

The non constant nature of the perimeter growth over time as suggested in Figure 4.7 have potential modeling implications. By stratifying the training data into day

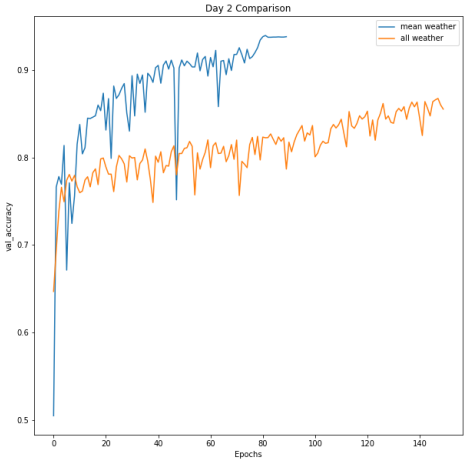
dependent samples and training unique models on each sample set, we see that the days with greater change in perimeter are much harder for the model to learn. Figure 4.8 shows the models converging fully on the second and third days of the wildfire. Alternatively, the final days from the training set are a much great challenge for the model. We see that the validation and the training learning curves are unstable and have not converged. Next we make an additional change to the input data.

4.5.2 Learning from Distributions of Input

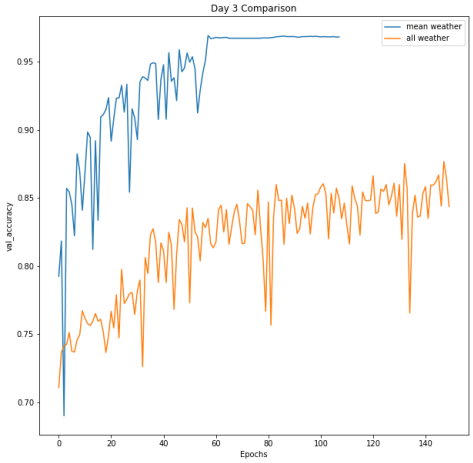
In previous results presented, the model was given access to the mean atmospheric conditions from 24 hours prior to the prediction time period. We suspect that the history at a higher resolution may provide useful additional information. As opposed to feeding the model a statistic that describes a time series distribution of input data, we input the full distribution.

We retrain the model with additional training data including a full tensor tracking 24 hourly updates on 7 atmospheric variables per training time segment.

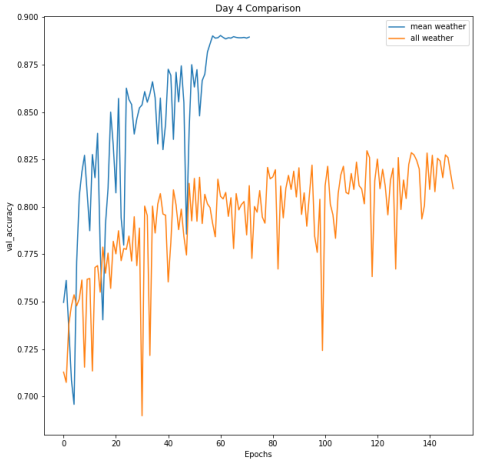
Holding the model itself constant, and only varying the aggregate atmospheric input data between daily mean and hourly distribution, we find some interesting results. The results of this alteration are displayed in Fig. 4.9.



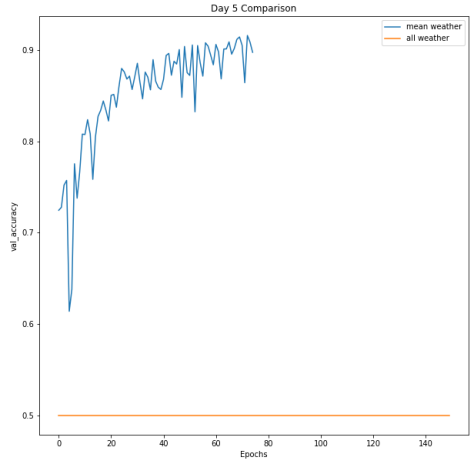
(a) Day 2



(b) Day 3



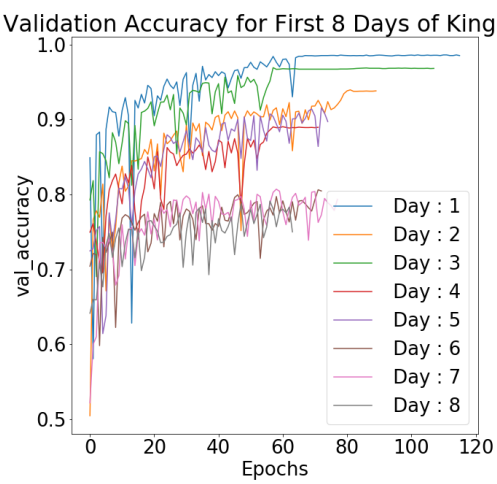
(c) Day 4



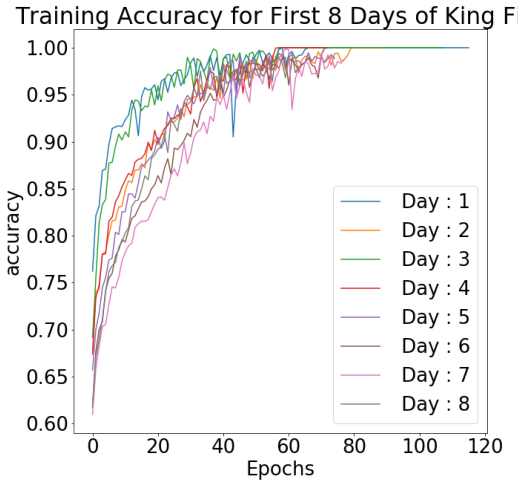
(d) Day 5

Figure 4.9: In each of the sub experiments, that the distributional tensor input of the atmospheric time-series only harms validation accuracy. Training is stopped when there is no longer changes in the validation loss after 25 consecutive epochs.

The learning curves from adding the all weather tensor to the training examples indicate that the model is perhaps suffering from a lack of tunable parameters with



(a) With added tunable parameters, all 8 days sampled from the training space converge to 100%



(b) All validation accuracies converge within 75%-98%

the additional data. We hypothesise that the additional data added to the training set will require a more complex model. To test this hypothesis we add additional dense layers after the weather tensor is concatenated to the convectional output. The result of this change is presented in Fig. 4.10a and Fig. 4.10b.

After adding additional layers to the model, we see a significant change in training and validation performance. The accuracies are bounded between 0.75% and 98%. By changing one component of the model or input data at a time, we improved weaknesses to produce quality intermediate results. The results displayed in 4.10a were also a result of tuning the learning rate and drop out probabilities using a grid search. The learning rate associated with the greatest performance was 0.00001 using the Adam gradient descent optimization algorithm. Further, the dropout was set to 0.3.

It is hard to compare the performance of this method with the reported performance of alternative methods as the input data and output are different. However, as

far as classifying pixels correctly over a 24 hour period, DeepFire shows comparable performance to the FireCast algorithm.

Modeling wildfires far into the future is a challenge for nearly every model, fully physical or data driven. The performance of FireCast, the methods discussed in Crowley, et all 2018 [34] and the benchmark physical models such as Farsite [18] all degrade over time. Producing accurate long term forecasts is one of the greatest challenge's in this deep learning task.

4.6 Conclusions, Limitations and Future Work

In this section, we described a model currently in development that predicts the evolution of The King Fire front in 24 hour segments. In addition to the structure and performance of the model we also discussed the curation of a large atmospheric and environmental data set that is used to train models. The data set currently contains the geo-referenced LiDAR, spectral layers, fire perimeter map, atmospheric data and ground cover maps,for nine major fires in the California Sierra Nevada's that occurred between 2010 and 2017. The current fires represented include the King, Cascade, Cedar, Rocky, Stone, Tubbs, Camp, Redding and Kincade Fire. These instances account for 102 days of burning fire and over 900,000 burned acres. The functional DeepFire API can be used to collect additional wildfire data from anywhere within the United States.

The model that is produced by this work is able to predict a wildfires development over a 24 hour period with up to 95% validation accuracy. This accuracy decays as

the model makes further future predictions and is not robust to all 24 hour windows.

Model Assumptions and Biases

Like many models, DeepFire holds many physical and theoretic assumptions about the incoming data and how it may represent reality. In addition to underlying assumptions, we produce several instances of bias through the data pre-processing and collection step.

We assume that the spectral layers accurately represent the vegetal landscape, that fire spreads asynchronously and that the local weather system behaves independently from the fire itself. It has been shown that in fact there can be positive feedback loops between the local atmosphere and heat from a wildfire. This in turn can influence how the fire will spread. One benefit of a black box model, is that we are not attempting to make any inference about the relationship between the input parameters and the response variable. In coupled fully physical models of wildfire spread, processes like advection and heat transfer are idealized. Idealization can be a convenient way to model something in uniform conditions, however its a strong assumption to think that idealizations will produce accurate results during realistic non uniform conditions. This is particularly true of non linear systems that exhibit a sensitive dependence on initial condition, which can be true of wildfire spread under realistic conditions [74].

Additionally, there are several elements of bias present in this analysis. Foremost, the treatment of the landscape sample space biases sampling. A birds eye view snapshot of earth represents a subset of the globe. If there is a fire spreading within the bounds of the snapshot, the relative area within the snapshot that the fire occupies is dependent on the bounds. Larger bounds produces a smaller relative area. The

original bounds of the satellite image are anywhere from 100-400 square miles in area depending on the position of the fire within one to four raster tiles. As our data collection method involves randomly sampling from a raster tile, the class balance is heavily effected by the size of the raster with respect to the size of the fire. We choose to limit the size of the processed rasters based on the size of the largest extent of a given fire event. This biases the model towards getting more examples of fire during sampling, than sampling the uncropped raster. In this way, we introduce some oversampling bias.

A fundamental problem with these types of black box models are edge cases. It is possible that there are many wildfire dynamics that happen in reality that are not contained within the training set. These types of anomalies will likely not be handled well by the model. Additionally, there may be assumptions made during the collection raw collection process that the authors of this work may not be aware of.

Future Work

The primary weakness of the final model is the ability to forecast further into the future than 24-48 hours. Thus, the focus of further research will be on using a more complex model to try and learn an explicit temporal embedding. Specifically, we plan to use Long Short Term Memory (LSTM) blocks to help the model learn the sequential nature of the fire progression. Adding a temporal encoding changes the fundamental treatment of this problem as a Markov chain.

Further the use of a binary mask output describing the future states of more than one location in the grid is an additional logical next step. This type of mask output is used in segmentation tasks such as cancer detection as exampled in the work of

Sirinukunwattana et al [75].

Additionally, adding explicit grid cell level information like land cover class, and the ability for the model represent extinguishing fire are two future improvements. The raw, preproccessed and cleaned data set is currently being hosted on a gitlab repository which can be accessed at the following URL : https://gitlab.com/maxfieldeland/deep_fire_data. The full API wrapper is also hosted on the repository.

Bibliography

[1] Mark A Finney. Fire growth using minimum travel time methods. *Canadian Journal of Forest Research*, 32(8):1420–1424, 2002.

[2] Aldo Starker Leopold. *Wildlife management in the national parks.* US National Park Service, 1963.

[3] Carol Miller and Dean L Urban. Modeling the effects of fire management alternatives on sierra nevada mixed-conifer forests. *Ecological Applications*, 10(1):85–94, 2000.

[4] Jan W Van Wagtendonk. The history and evolution of wildland fire use. *Fire Ecology*, 3(2):3–17, 2007.

[5] Omer Call Stewart. *Forgotten fires: Native Americans and the transient wilderness.* University of Oklahoma Press, 2002.

[6] Jennifer R Marlon, Patrick J Bartlein, Daniel G Gavin, Colin J Long, R Scott Anderson, Christy E Briles, Kendrick J Brown, Daniele Colombaroli, Douglas J Hallett, Mitchell J Power, et al. Long-term perspective on wildfires in the western usa. *Proceedings of the National Academy of Sciences*, 109(9):E535–E543, 2012.

[7] Kevin C Ryan, Kristine M Lee, Matthew G Rollins, Zhiliang Zhu, James Smith, and Darren Johnson. Landfire: Landscape fire and resource management planning tools project. In *In: Andrews, Patricia L.; Butler, Bret W., comps. 2006. Fuels Management-How to Measure Success: Conference Proceedings. 28-30 March 2006; Portland, OR. Proceedings RMRS-P-41. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. p. 193-200*, volume 41, 2006.

[8] John T Koehler. Prescribed burning: a wildfire prevention tool. *Fire Management Notes*, 53(54):9–13, 1993.

[9] Mollie Kathleen Buckland. What is a megafire? defining the social and physical dimensions of extreme us wildfires (1988-2014), 2019.

[10] J Williams and L Hamilton. The mega-fire phenomenon: Toward a more effective management model–a concept paper. the brookings institution. *Center for Public Policy Education, Washington, DC, 19p*, 2005.

[11] Susan Prichard, N Sim Larkin, Roger Ottmar, Nancy HF French, Kirk Baker, Tim Brown, Craig Clements, Matt Dickinson, Andrew Hudak, Adam Kochanski, et al. The fire and smoke model evaluation experiment – a plan for integrated, large fire–atmosphere field campaigns. *Atmosphere*, 10(2):66, 2019.

[12] 2019 verisk wildfire risk analysis: Property underwriting.

[13] Emily Jane Davis, Cassandra Moseley, Max Nielsen-Pincus, and Pamela J Jakes. The community economic impacts of large wildfires: A case study from trinity county, california. *Society & Natural Resources*, 27(9):983–993, 2014.

[14] Anatoli Grishin. *Mathematical modeling of forest fires and new methods of fighting them*. Publishing house of the Tomsk state university, 1997.

[15] Richard C Rothermel. A mathematical model for predicting fire spread in wildland fuels. *Res. Pap. INT-115. Ogden, UT: US Department of Agriculture, Intermountain Forest and Range Experiment Station. 40 p.*, 115, 1972.

[16] M Larini, F Giroud, B Porterie, and J-C Loraud. A multiphase formulation for fire propagation in heterogeneous combustible media. *International Journal of Heat and Mass Transfer*, 41(6-7):881–897, 1998.

[17] Ian Knight and John Coleman. A fire perimeter expansion algorithm-based on huygens wavelet propagation. *International Journal of Wildland Fire*, 3(2):73–84, 1993.

[18] Mark A Finney. Farsite: Fire area simulator-model development and evaluation. *Res. Pap. RMRS-RP-4, Revised 2004. Ogden, UT: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. 47 p.*, 4, 1998.

[19] Rodman Linn, Jon Reisner, J. Coleman, and S. SMITH. Studying wildfire behavior using firetec. *International Journal of Wildland Fire*, 11, 11 2002.

[20] Mark A Finney. An overview of flammap fire modeling capabilities. In *In: Andrews, Patricia L.;Butler, Bret W., comps. 2006. Fuels Management-How to Measure Success: Conference Proceedings. 28-30 March 2006; Portland, OR. Proceedings RMRS-P-41. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. p. 213-220*, volume 41, 2006.

[21] Jan Mandel, Jonathan D Beezley, Janice L Coen, and Minjeong Kim. Data assimilation for wildland fires. *IEEE Control Systems Magazine*, 29(3):47–65, 2009.

[22] Jan Mandel, Jonathan D Beezley, and Adam K Kochanski. Coupled atmosphere-wildland fire modeling with wrf-fire. *arXiv preprint arXiv:1102.1343*, 2011.

[23] O. SĂŠro-Guillaume and J. Margerit. Modelling forest fires. part i: a complete set of equations derived by extended irreversible thermodynamics. *International Journal of Heat and Mass Transfer*, 45(8):1705–1722, 2002.

[24] Andrew L Sullivan. Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models. *International Journal of Wildland Fire*, 18(4):387–403, 2009.

[25] Colomba Di Blasi. Modeling and simulation of combustion processes of charring and non-charring solid fuels. *Progress in energy and combustion science*, 19(1):71–104, 1993.

[26] RO Weber. Toward a comprehensive wildlife spread model. *International Journal of Wildland Fire*, 1(4):245–248, 1991.

[27] Dougal Drysdale. *An introduction to fire dynamics*. John Wiley & Sons, 2011.

[28] Jean-Francois Sacadura. Radiative heat transfer in fire safety science. In *ICHMT DIGITAL LIBRARY ONLINE*. Begel House Inc., 2004.

[29] F Furcas, A Zucca, DL Marchisio, and AA Barresi. Mathematical modelling of soot nanoparticles formation and evolution in turbulent flames. In *Proceedings for the 30th Meeting of Combustion*, 2007.

[30] Sarah Holder and CityLab. Playing the odds on the next california wildfire, Feb 2020.

[31] L Ferragut, MI Asensio, and S Monedero. A numerical method for solving convection-reaction-diffusion multivalued equations in fire spread modelling. *Advances in Engineering Software*, 38(6):366–371, 2007.

[32] J Margerit and O SĂŠro-Guillaume. Modelling forest fires. part ii: reduction to two-dimensional models and simulation of propagation. *International Journal of Heat and Mass Transfer*, 45(8):1723 – 1737, 2002.

[33] David Radke, Anna Hessler, and Dan Ellsworth. Firecast: leveraging deep learning to predict wildfire spread. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4575–4581. AAAI Press, 2019.

[34] Sriram Ganapathi Subramanian and Mark Crowley. Using spatial reinforcement learning to build forest wildfire dynamics models from satellite images. *Frontiers in ICT*, 5:6, 2018.

[35] William E Mell, Samuel L Manzello, Alexander Maranghides, David Butry, and Ronald G Rehm. The wildland–urban interface fire problem–current approaches and research needs. *International Journal of Wildland Fire*, 19(2):238–251, 2010.

[36] Patricia L Andrews. Behaveplus fire modeling system: past, present, and future. In *In: Proceedings of 7th Symposium on Fire and Forest Meteorology; 23-25 October 2007, Bar Harbor, Maine. Boston, MA: American Meteorological Society. 13 p.*, 2007.

[37] Ana CL Sá, Akli Benali, Paulo M Fernandes, Renata MS Pinto, Ricardo M Trigo, Michele Salis, Ana Russo, Sonia Jerez, Pedro MM Soares, Wilfrid Schroeder, et al. Evaluating fire growth simulations using satellite active fire data. *Remote Sensing of Environment*, 190:302–317, 2017.

[38] NOAA. Home: Noaa physical sciences laboratory.

[39] Svante Arrhenius. Uber die Reaktionsgeschwindigkeit bei der Inversion von Rohrzucker durch Sauren, January 1889.

[40] GN Mercer and RO Weber. Combustion wave speed. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 450(1938):193–198, 1995.

[41] Geoff N Mercer, HS Sidhu, RO Weber, and V Gubernov. Evans function stability of combustion waves. *SIAM Journal on Applied Mathematics*, 63(4):1259–1275, 2003.

[42] Robert E Burgan. *Behave: fire behavior prediction and fuel modeling system, fuel subsystem*, volume 167. Intermountain Forest and Range Experiment Station, Forest Service, US, 1984.

[43] Zhong Zheng, Wei Huang, Songnian Li, and Yongnian Zeng. Forest fire spread simulating model using cellular automaton with extreme learning machine. *Ecological Modelling*, 348:33–43, 2017.

[44] Ioannis Karafyllidis and Adonios Thanailakis. A model for predicting forest fire spreading using cellular automata. *Ecological Modelling*, 99(1):87–97, 1997.

[45] Xiaolin Hu, Yi Sun, and Lewis Ntaimo. Devs-fire: design and application of formal discrete event wildfire spread and suppression models. *Simulation*, 88(3):259–279, 2012.

[46] François Rebaudo, Verónica Crespo-Pérez, Jean-François Silvain, and Olivier Dangles. Agent-based modeling of human-induced spread of invasive species in agricultural landscapes: insights from the potato moth in ecuador. *Journal of Artificial Societies and Social Simulation*, 14(3):7, 2011.

[47] Dirk Helbing. Agent-based modeling. In *Social self-organization*, pages 25–70. Springer, 2012.

[48] Jinghui Zhong, Linbo Luo, Wentong Cai, and Michael Lees. Automatic rule identification for agent-based crowd models through gene expression programming. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1125–1132. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[49] Mauro Castelli, Leonardo Vanneschi, and Ales Popovic. Predicting burned areas of forest fires: An artificial intelligence approach. *Fire Ecology*, 11:106–118, 04 2015.

[50] Steven M Manson. Bounded rationality in agent-based models: experiments with evolutionary programs. *International Journal of Geographical Information Science*, 20(9):991–1012, 2006.

[51] Steven M Manson. Agent-based modeling and genetic programming for modeling land change in the southern yucatan peninsular region of mexico. *Agriculture, ecosystems & environment*, 111(1-4):47–62, 2005.

[52] Tommaso Toffoli and Norman Margolus. *Cellular automata machines: a new environment for modeling*. MIT press, 1987.

[53] János Neumann, Arthur W Burks, et al. *Theory of self-reproducing automata*, volume 1102024. University of Illinois Press Urbana, 1966.

[54] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.

[55] Ken Perlin. Improving noise. In *ACM transactions on graphics (TOG)*, volume 21, pages 681–682. ACM, 2002.

[56] Ken Perlin. Improving noise. *ACM Transactions on Graphics*, 21:681–682, 2002.

[57] Kerman Phillip. *Macromedia Flash 8 @work, Projects and Techniques to Get the Job Done*. Sam's Publishing, 2006.

[58] John R Koza et al. *Genetic programming II*, volume 17. MIT press Cambridge, 1994.

[59] De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, Christian Gagné, et al. Deap: A python framework for evolutionary algorithms. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 85–92. ACM, 2012.

[60] Vipul K Dabhi and Sanjay Chaudhary. A survey on techniques of improving generalization ability of genetic programming solutions. *arXiv preprint arXiv:1211.1119*, 2012.

[61] Alexandre Bonnet and Moulay A Akhloufi. Uav pursuit using reinforcement learning. In *Unmanned Systems Technology XXI*, volume 11021, page 1102109. International Society for Optics and Photonics, 2019.

[62] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu. Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2894–2902, 2016.

[63] Justin J Podur and David L Martell. The influence of weather and fuel type on the fuel composition of the area burned by forest fires in ontario, 1996–2006. *Ecological Applications*, 19(5):1246–1252, 2009.

[64] The national map: https://viewer.nationalmap.gov/advanced-viewer/.

[65] Climate Change Canada. Government of canada, Jul 2019.

[66] Eduardo Rocha Rodrigues, Igor Oliveira, Renato Cunha, and Marco Netto. Deepdownscale: a deep learning strategy for high-resolution weather forecast. In *2018 IEEE 14th International Conference on e-Science (e-Science)*, pages 415–422. IEEE, 2018.

[67] Jarrett Booz, Wei Yu, Guobin Xu, David Griffith, and Nada Golmie. A deep learning-based weather forecast system for data volume and recency analysis. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 697–701. IEEE, 2019.

[68] Szu-Yin Lin, Chi-Chun Chiang, Jung-Bin Li, Zih-Siang Hung, and Kuo-Ming Chao. Dynamic fine-tuning stacked auto-encoder neural network for weather forecast. *Future Generation Computer Systems*, 89:446–454, 2018.

[69] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *arXiv*, pages arXiv–2003, 2020.

[70] Toby N Carlson, Robert R Gillies, and Eileen M Perry. A method to make use of thermal infrared temperature and ndvi measurements to infer surface soil water content and fractional vegetation cover. *Remote sensing reviews*, 9(1-2):161–173, 1994.

[71] SC Nayak, Bijan B Misra, and Himansu Sekhar Behera. Impact of data normalization on stock index forecasting. *Int. J. Comp. Inf. Syst. Ind. Manag. Appl*, 6:357–369, 2014.

[72] François Chollet et al. Keras. https://keras.io, 2015.

[73] Geomac database.

[74] Mélanie C Rochoux, Sophie Ricci, Didier Lucor, Bénédicte Cuenot, and Arnaud Trouvé. Towards predictive data-driven simulations of wildfire spread–part i: Reduced-cost ensemble kalman filter based on a polynomial chaos surrogate model for parameter estimation, 2014.

[75] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging*, 35(5):1196–1206, 2016.