Using Spark Simulations in Drone Navigation for Reducing Uncertainty when Fighting Wildfires*

Dr. Martin Aleksandrov

Freie Universitat Berlin

Abstract.

Wildfires are a common issue for hundreds of millions of years and they often cause massive damage. Because wildfire can be caused both by humans, partly intentionally, partly unintentionally, and by recurring natural catastrophes, such as storms, it is not foreseeable that forest fires will disappear any time soon. Nevertheless, mankind is making efforts to prevent both the occurrence of forest fires and their uncontrolled spread. Part of this effort is applying and linking new scientific knowledge with proven means of fighting wildfires. Artificial intelligence and machine learning are playing an increasingly important part because they enable more accurate and resources efficient ways to predict wildfire. In this report we are going from simulating a wildfire, over analysing data to the use of algorithm for smart allocating of firefighters to fight wildfire.

Keywords: Wildfire · Drones · Machine Learning

1 Introduction (Hypotheses, Aims)

Wildfires aren't limited to particular ecosystems and environments and can, if the Topography plays into their hand grow rapidly. While wildfires often help keeping ecosystems healthy, they also threaten the lives of humans and can cause massive damage to agriculture and infrastructure. During the summer 2019 to 2020 Australia was hit by the most catastrophic bushfire season. Up to 19 million hectares were burnt [1]. Worldwide wildfires are about to get more frequent and intense, with a global increase of extreme fires of up to 14 per cent by 2030 and 30 per cent by the end of 2050 according to a report by UNEP and GRID-Arendal [2]. Australia and the rest of the world get prepared for wildfires. Therefor scientific papers had been published to measure damage, like the Article "Environmental circumstances surrounding bushfire fatalities in Australia 1901-2011" [3] which analysis a dataset covering bushfire related life loss in Australia and to explore countermeasures, like "Probabilistic allocation and scheduling of multiple resources for emergency operations" [4], a paper presenting an emergency operation model that aims to facilitate the scheduling and sequencing resources using multiple stochastic scenarios. Wildfires are made worse by climate change through increased drought, high air temperatures, low relative humidity, lightning, and strong winds resulting in hotter, drier, and longer fire seasons. To cope this factors and to analyse their direct impact on wildfires, we took the approach of analysing handful of this factors impacts at a single wildfire, placed in Australia. We're going to show in a detailed way how simulation leads to information gain. Through the data we collected it became possible to make a risk analysis and develop an efficient way to allocate drones near wildfires.

[1] WWF Report

[2] UNEP-Report

[3] Environmental circumstances surrounding bushfire fatalities in Australia 1901-2011

[4] Probabilistic allocation and scheduling of multiple resources for emergency operations; a Victorian bushfire case stu

2 Motivation

In the time span of our research during the semester, we worked with different scientific papers, with every paper increasing our motivation to deal with the concern of wildfires. We picked three papers to explain our state of knowledge and the fundamental of this report, every paper viewed a different angle on the working against the risk and damage of wildfire. In the first [5] Firefly was first introduced, a new resources allocation algorithm, which has a high impact on how we decided to look at the analysis of our data, and which way we took to implement machine learning Algorithm. It first formulates the drone allocation as maximal set coverage problem, and uses a greedy algorithm to allocate the drones. Then by using the observed and estimated data Firefly solves the firefighter allocation problem as a knapsack. In the process of fighting wildfires by allocating firefighters, we took a more basic approach and started by simulating fire and looking at the different properties of wildfires.But the guideline of first allocating observation drones to collect information and then allocate firefighting units, by evaluation of data in real-time, described in this paper also lead us through our research. With the second research paper [6] showing how to use social media to detect and locate wildfires. During a fixed research time the authors analyzed on the bases of three models if wildfires lead to a higher output on social media in the same or neighbouring regions. This phenomena was tested and overlooked in different US states, with a lot of states showing strong positive correlation and no state showing significant negative correlations. The third paper [7] took a more hands on approach and decided to help dealing with wildfires by taking the idea of sharing resources and developing a system to do the sharing more efficient. For the sharing process to work more efficient it takes three key components: The distressed agency, finding itself in a situation where its current level of resources is inadequate. The broker for resource allocation, receives request from Distressed Agency and broadcast the Request to the lending agencies. The lending agencies, balance their own resources needs with benefits and cost of lending some of their resources to distressed agency. All three have to work together and make strategic decision, when making resources available, also they have have to deal with lots of uncertainties in the calculation and what leads back to our first paper, information about the strength and the speed of the fire are key factors in good decision making. To get those information, simulating wildfires are one way to get a lot of information. One tool we used for simulation, will be more specific explained in this report, but there is also a different tool, which should be mentioned: Wildfire Analyst Pocket, a tool, which works with the following regulators: Fuel, Wind, Wind Dir, Slope, Aspect , Air Temperature, Relative Humidity and Live Moist. Similar Parameters like Spark and also enables us through different graphs to generate an possible output for us to analyze and afterwards conclude different measure to meet wildfires.

[5] Fighting Wildfires under Uncertainty:A Sequential Resource Allocation Approach
[6] Using Social Media to Detect and Locate Wildfires
[7] Resource Sharing for Control of Wildland Fires

3 Related Work

The paper [8]addresses the terms ”Fire Intensity,” ”Fire Severity,” and ”Burn Severity,” which are often confused or used interchangeably, but are actually different concepts in the context of wild- fires. In the paper, Keeley clarifies that Fire Intensity is the rate at which energy is released per unit time while a fire is burning. Fire Severity refers to the effects of fire on vegetation and soil, such as killing trees or damaging soil. Burn Severity refers to the impact of the fire on the soil, such as the depth and extent of the burn. Keeley emphasizes that a clear distinction between these terms is important to better understand wildfire impacts and make more effective management decisions. In terms of use, Keeley recommends that Fire Intensity and Fire Severity be used to describe fires in terms of their ecological impacts, while Burn Severity should be used to describe fires in terms of their ground impacts. Overall, this paper has been very helpful in expanding our understanding and giving a consistent definition of the most important terms in a wildfire. The following paper has inspired us in calculating the cost of a fire and the resulting damage[9].More specifically, it deals with the development of a mathematical model to optimize resource allocation in wildfire mitigation. The authors use an integer programming model that considers several variables, such as the number of responders, fire suppression strategy, and geographic conditions, to determine the best resource allocation. The results show that the model can help improve the effectiveness and efficiency of resource allocation in firefighting.

4 Main Body

4.1 Spark

Spark is a wildfire simulation toolkit created by the research Team of CSIRO Research and enables to visualize possible and already happened wildfires around the world and to output the information gained through those simulation in a variety of data formats.

Environment Wildfire behaviour differentiate strongly when the environments it occurs differs. Spark differentiate between at least four habitats. There are areas where its impossible to ignite a fire and already existing wildfire stop. Areas where fire itself burn quiet well, reoccurring areas in this research to wildfires were grass- and bush lands. Areas infrastructure is build which have a higher value flames still gain momentum in this areas, but aren’t expanding as quick as in forest for example and not fire friendly areas which slow the spread of fire, areas which may are separated by rivers or have not forest friendly mountains. Spark also differentiate between Layers, visible in Initialisation >Initialisation start

Spark-GUI In this section we describe the functionality of the Spark-GUI, we were able to download for free and the different changes in modification we took to get to the simulation of a wildfire, which we gained our data from.
The Spark-GUI provides the following configurations:
Viewer, Configuration, Data input, Series input, Initialization, Rate of Speed, Post-processing, Experimental, Log
In the Process of Simulating Wildfires we used options in View, Configuration, Series input, Post-Processing, Experimental and Log.

[8] Fire intensity, fire severity and burn severity: a brief review and suggested usage
[9] An Integer Programming Model to Optimize Resource Allocation for Wildfire Containment

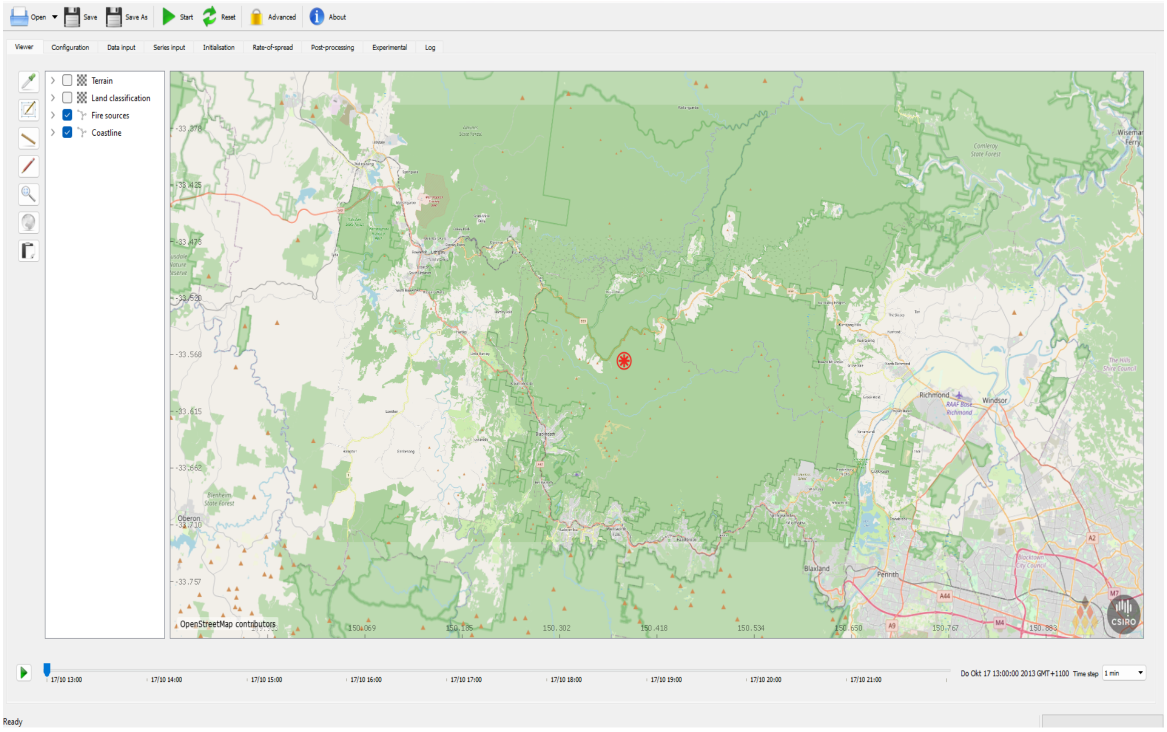

Fig. 1. Spark-GUI

Viewer Enables map overview with different options to manipulate and change the layers, adding geometric figures and measure distances. It's also possible to change the layer in other ways adding layers of calamities and surrounding of the fire. The Viewer also shows the way fire changes its course and what areas are met by the fire storm. Its based on OpenStreetMap and can be used to show different fires at the same time and even located in different continents. We create different ignition points and are able to see over our a prior chosen time-period how those fires spread over the landscape. In the end frame there are different possible outputs. In this case we see a coloured fire, which shows the points in time the fire reaches different stretches of land.

Configuration Here most of the configuration happens and its possible to chose the output files. In Configuration: We select the timespans the starting points and the name of our project. It's also shown where the fire sources lay, the python script to start fire and the different output files we plot our data into. We can add geometric figures to the layout of our fire, see information about the ignition sources and other geometric points displayed in the map.

Data input The category Data input has different childhood categories we work with to change the spread of fire, first the Meteorological inputs, here changes of weather are possible, which impact the spread and the ignition of almost every wildfire. It subcategories are differentiated between:

- Source file names, is used to add mask, which changes the behavior of parameters, which we're going to analyze in more detail
- Source directory, enables adding layers
- Transform, applies scale and offset values to each gridded value.

Other child categories are:

- Terrain inputs, is used to change the typography. Important are the different soil conditions, on the website of spark it's possible to find Plugins. Because it's a factor, which differs by every change of location and has a high impact on the expanding of fire.
- Custom inputs

Additionally another option was the Gridded bounds where it's possible to define a fixed framework we review and manipulate our fire in. Most interesting was the Meteorological inputs Transform there it is possible to change meteorological parameters like wind directions wind speed Temperature and relative humidity. There exist two ways of impacting this criterion through data input whether changing the directories and files which have impact or through directly scaling the specification in Spark.

Series input One of most used sections is the script file, a file where it's possible to see the different input parameter in excel and pick a specific moment in time where to change them. We reviewed different graphs as part of the mechanism of analyse conditions and fire influencing factors. Spark shows detailed graphs of the changes of Wind Speed, Wind direction, Air Temperature, Relative humidity, Drought Factor, Fuel State during the simulation.

4.2 Log

Spark Log became an important segment in the process of collecting data. We get an impression how the fire evolves over time and we get in the documentation of these changes ”timestamps”. Next to the natural function of an Log-file to protocol and document changes and information appearing in an process the Log in Spark covers in the window ”simulation log” the detailed development of our fire during and after the simulation. The exact information in the window are categorized in different groups.

- First defined is the start and end time.
- Second the different stamps where we see moments the fire expanse, the stamps are moment in time and fire expanse** is covered through area in percentage is shown how much of the simulation work is already done.
- Third is the total time the execution took
- Fourth the output files the results get written into
- Fifth is an list of information with the output parameters in the fire.

The Average speed in m/s, the maximum intensity in kW/m, the maximum flame height in m, the covered fire area in ha and the estimated perimeter in km. In the process we started with a base case simulation, which run almost unchanged to the packed fire project spark enables us to use.

4.3 Simulations

After this Base Case changing the following parameters

- Precipitation,
- Air Temperature,
- Relative Humidity,
- Wind spreed,
- Wind direction

enables us to simulate total different outcomes. Those confines are clustered as input parameters. Spread over different functions in Spark mainly Series input >Script file. It allows to manipulate parameters at any given point in time, hence it becomes possible to counterfeit changing weather conditions. In the first simulations we constantly change the timespan and the ignition points. We settled for one ignition point at the coordinates [150.378, -33.5724]and as time interval we decide to go with five days and nine hours seated in the middle of October. It allows the fire to grow at an exceptional rate and in extreme cases let it reaches bigger accumulation of infrastructure and villages. In the process we got a variation of results, without exceeding our computing capacities. During the simulation Spark allows to stop and look at graphs which show data output to specific point in time. The graphs are to be find in the point Series input. Also its possible to add geometric figures into viewer which are able to change the development of fire. With this opportunities we created a solid data set, which in total consists of 41 Instances, one Instance consists of the mentioned five input parameters, which change on a regular bases during the simulation. This changes are documented through the graphs in Spark and six output parameters which we were able to read out of log and output files:

- Area,
- Maximum flame Height,
- Maximum flame intensity,
- Estimated perimeter,
- Average speed,
- Fire Area

4.4 Data analysis

Open Source Software and implemented in Java. Weka, including the early non-Java predecessors of Weka 3, was developed at the Department of Computer Science of the University of Waikato in Hamilton, New Zealand. Weka is highly spread in academic and business communities. Makes it possible, by uploading the collected data from Spark to analyse and visualize this data. Weka is developed for machine learning and provides a wide range of tools. Our first step into the Data-analysis was to get a view on how the parameters works together and which data can be used to write a risk analysis and to allocate firefighters. Going through the following menu points: Weka Explorer >Preprocess >Open file >Visualize All allows to visualize our Attributes in ranges of their distribution. There it is possible to have a first overview over the data and to see differences. In Weka the interactive graphic is displayed in a different window, which gives the possibility to work in detail with the gained information.

We differentiate between input parameters, which distribution is explained through the fact that for generating data points their values were changed repeatedly. Every parameter is shown in the same graphics, but we look into Precipitation in mm, Air Temperature in degrees Celsius, Relative Humidity in percentage, Wind-Speed in km/h and Wind-Direction in degrees true in a different way then at Area in nectar, Average speed in m/s, Maximum Intensity in kw/m, Flame height in m, Perimeter in km and Fire Area in nectar. Even if their datatype differs, for example between Air Temperature in C and Wind-Speed in m/s all values are classified as numeric therefor it's easy for us to evaluate those data and compare them: Overview over distribution: Precipitation [0, 1400]; Air Temperature [7, 50]; Relative Humidity [10, 90]; Wind-Speed [22, 200]; Wind-Direction [20, 310]; Output Parameter: Area[270, 187237]; Average-Speed[0, 0,2]; Maximum-Intensity[95, 353807.42]; Flame height[5.19, 101177.3]; Perimeter[6.34, 80925]; Fire Area[0, 270303.36] In the following we want to take a closer look at some of our Output Parameters: Area, Average Speed and Maximum Intensity This Reprocessed Data, gives us a first impression of the value of the data we accumulated, if the high and lower bounds are near to each other. It directly represents result of different simulation
A deeper look into Area:

- Area is the amount of land were an active fire burned, therefor every hectare in the amount took some damage, from firefighting point of view the best case scenario is equal to the minimum scenario.
  In the minimum case the fire covers a range between 270 and 31431.167 ha
  Twenty two and accordingly most of the simulated fires are seated between 31431.168 and 62592.33 ha
  One fire covered a bigger area, high runner which is seated between 156075.833 and 187237 ha Around eleven times the area size of the medium of the lowest bound. This shows the danger of wildfires, because the same spark can under changing environmental circumstances lead to a far more devastating fire, there accurate and mistake-free simulation are from high importance.
  Mean is 64801.971 ha

- Average Speed:
  Minimum in range between [0, 0.041]
  Highest in range between [0.163, 0. 203]
  Mean is 0.097
  Both Types correlate strongly, in graphs this information is not clearly visible.

- Maximum Intensity:
  Minimum in range between [95, 70837.83]
  Highest in range between [283064.952, 353 807.406]
  Mean is 107072.336
  There isn't a clear and visible correlation between Maximum Intensity and the two parameters Average Speed and Area, therefor it's hard to compare.

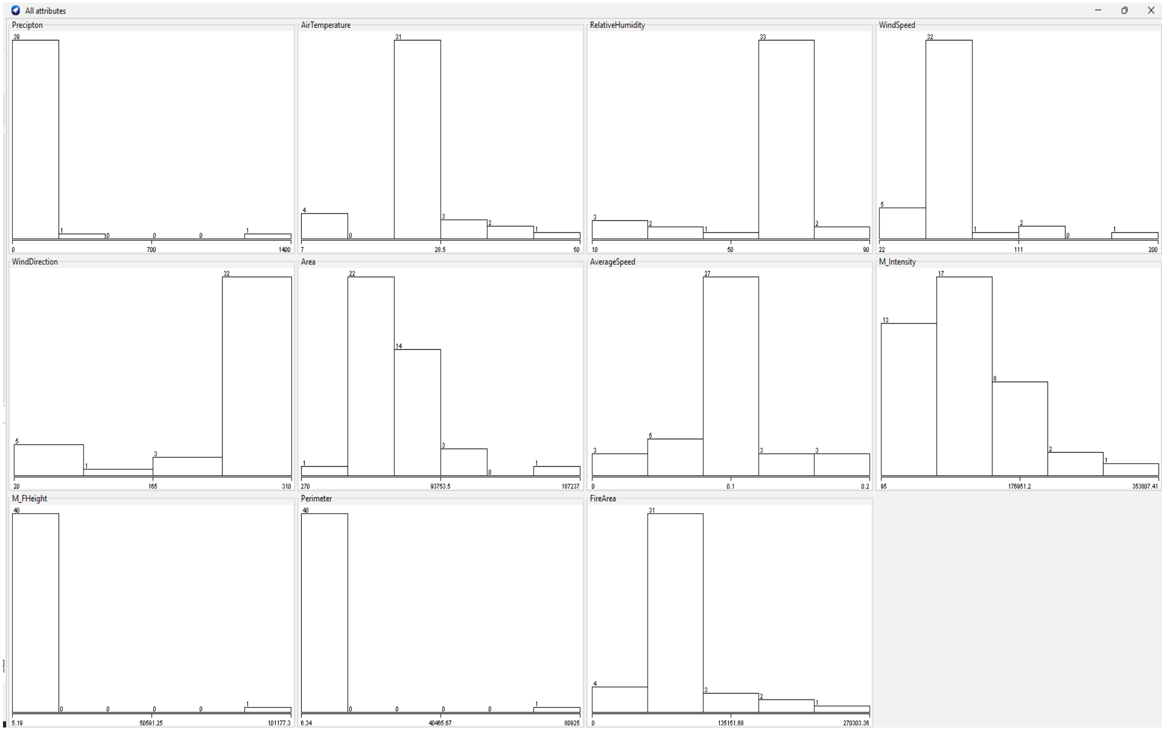

Fig. 2. Distribution through graphs

Overview Parameters distribution:

- Precipitation 39, 1, 0, 0, 0, 1
- Air-Temperature 4, 0, 31, 3, 2, 1
- Relative Humidity 3, 2, 1, 33, 2
- Wind-Speed 5, 32, 1, 2, 0, 1
- Wind-Direction 5, 1, 3, 32
- Area 1, 22, 14, 3, 0, 1
- Average-Speed 3, 5, 27, 3, 3
- Maximum Intensity 13, 17, 8, 2, 1
- Flame-Height 40, 0, 0, 0, 0, 1
- Perimeter 40, 0, 0, 0, 0, 1
- Fire-Area 4, 31, 3, 2, 1

In the next step, we glance further into the correlations of our Data. First we mark the correlations between our input Parameters. Their starting values are changed manually and they don't hold information about the simulated fire, to understand how those parameters interact and influence each other is important for us, because it helps to calculate and predict fire development. Correlation is measured between [-1, 1] with every value smaller 0 is a called "negative correlation" meaning they still have impact on each other. E.g.: A high Wind Speed negates the impact of high Air Temperature. Most interesting for us would be a correlation of 1, also called perfect correlation. A correlation of -1 is a perfect negative correlation meaning the values completely canceling each other out. A Correlation near 0 or 0 is weak to not existing correlation. A look at every input parameter and their highest and lowest correlations: Precipitation has its high and strongest correlation with Wind Direction about 0.0866 and the correlation closest to 0 is with Air Temperature about -0.0115. Recognizable is that Perception has a majority of negative correlations.

Air Temperature: Highest and strongest correlation with Wind Direction 0.33618 Weakest with Precipitation -0.01146 Lowest with Wind Speed -0.05316 Relative Humidity High and strongest correlation with Air Temperature 0.15797 Lowest and weakest correlation with Wind Direction -0.00402 Wind Speed High and strongest Correlation with Wind Direction 0.0909 Lowest and weakest with Precipitation with -0.0483 As we can see every input Parameter influences also other input parameters. This is common and explained by fire is to be viewed as one element and everything what influences fire also influences other elements. More interesting for risk analysis is the the idea of looking at relation between input and output parameters. While we can expect the correlation between the output parameters to be random and don't help our analysis. They are fixed parameters which represent the expand of the fire. Correlation between input and output parameters: Interesting in this context is which input parameter is most relatable for one output parameter. To analyze this cases most efficient we created so called correlation matrix where we clearly see our input parameters influences. Correlation Matrix is an efficient way to gain overview over a lot of correlations with more different parameter. In our case we, are we showing around 11 parameters in the correlation. In this case we look at the Parameter Area, which correlates with itself always with one. One side of the matrix is redundant because every correlation is already shown in one half. The strongest correlation we see is 0.79 a correlation near one. The weakest correlation we detect is -0.01. Therefor we see big differences in how values correlate with each other.

Area:

$$\begin{matrix}
1 & -0.01 & 0.06 & -0.05 & 0.09 & -0.08 & -0.15 & -0.03 & -0.03 & -0.06 \\
-0.01 & 1 & 0.16 & -0.05 & 0.34 & -0.04 & 0.1 & -0.01 & 0.01 & 0.09 \\
0.06 & 0.16 & 1 & 0.07 & -0 & -0.72 & -0.3 & 0.05 & 0.05 & -0.68 \\
-0.05 & -0.05 & 0.07 & 1 & 0.09 & 0.14 & 0.55 & -0.04 & -0.08 & -0.11 \\
0.09 & 0.34 & -0 & 0.09 & 1 & -0.04 & -0.18 & -0.11 & -0.36 & 0.05 \\
-0.08 & -0.04 & -0.72 & 0.14 & -0.04 & 1 & 0.69 & -0.41 & 0.03 & 0.79 \\
-0.15 & 0.1 & -0.3 & 0.55 & -0.18 & 0.69 & 1 & -0.25 & 0.1 & 0.35 \\
-0.03 & -0.01 & 0.05 & -0.04 & -0.11 & -0.41 & -0.25 & 1 & -0.03 & -0.27 \\
-0.03 & 0.01 & 0.05 & -0.08 & -0.36 & 0.03 & 0.1 & -0.03 & 1 & -0.26 \\
-0.06 & 0.09 & -0.68 & -0.11 & 0.05 & 0.79 & 0.35 & -0.27 & -0.26 & 1
\end{matrix}$$

$$CorrelationmatrixArea$$

While the 1 always indicates the correlation of the value with itself the values right to 1 show the most interesting and most relatable.

5 Clustering

After the collection and Analysing the Data, turning back to Spark and laying the groundwork for the machine learning algorithm required to cluster the map. Therefor it becomes more accessible to determine where a fire outbreak and to have a place the risk and damage calculation is. It was only necessary to determine the zones the fire outbreak took places and the zones where a risk calculation should take place. The zones and clustering is also necessary for the allocation of drones so an analyze of fire intensity gets possible. This is a further step in supporting firefighters and they enable to look at spread of fire and enable to gain information about fire-behaviour.

6 Prediction of Risk

Fighting wildfires is a major concern for firefighters and other responders around the world. The effects of wildfires can be devastating. Because they are often difficult to control, they spread far and wide, damaging land, wiping out animals and plants, and in the worst cases costing lives.
However, fighting forest fires is a very difficult task. The problem is often that the exact fire areas cannot be effectively localized. Thus, resources, which are limited anyway, cannot be distributed effectively. In recent years, drones have proven to be a helpful tool in fighting wildfires. With the help of drones, responders can quickly and effectively gather and visualize information about the progress of a forest fire, which helps improve firefighting efforts. The use of drones is becoming increasingly popular.
Nevertheless, there is still a lot of potential in improving this technology. In reality, it is often the case that drones are also limited as a resource. This is problematic because a drone can only view a certain radius and provide information about it. Thus it happens quickly that with huge fire areas not the whole fire area can be covered, but only small subareas of it. In an ideal scenario, there would be enough drones to cover the entire fire area. Thus, one would have information about the fire intensity in individual zones and could distribute resources based on this information so that the maximum damage is minimized as much as possible. Since in reality there are often not enough drones to cover all areas, a different strategy is needed to determine the fire intensity for zones that are not covered by drones.

In the following sections, we will discuss in more detail how drones and machine learning can be used to fight wildfires. Here, we try to predict how a forest fire will develop based on data we have from simulation and eventually predict which zones will do the most damage. These zones should then be fought first to keep the damage as low as possible.

When deciding which algorithms to take for machine learning, we had to consider a few things. On the one hand, we have not yet collected too much data that we could take for training. Therefore, we need algorithms that do not require amounts of data to work well. Another challenge was that we have a multidimensional output. Although in the end it is mainly the risk that matters to us, we still need the values of other parameters that are crucial for the risk. Because our risk for a zone i is calculated by the following formula:

$$risk_{expected} = 0,3 * FlameHeight_i + 0,8 * \frac{1}{ArrivalTime_i} \tag{1}$$

As you can see, the risk is composed of two parts. The FlameHeight and the ArrivalTime play a role. FlameHeight is the height of the flame and gives us information about the fire strength of a zone. ArrivalTIme is the time it takes for a fire to reach a zone. We decided to use a weight of 0.3 for FlameHeight and a weight of 0.8 for ArrivalTIme. Our assumption was that we have expert knowledge and know these weights exactly. In further work, one could really consult an expert to determine these weights. Likewise, we could collect more data from simulations and use further analysis to identify these weights more precisely. We divide the ArrivalTime by one to give a shorter arrival time a higher weight.
The proposed formula is not final and will be extended in future work. Our goal is to extend the formula with more parameters. Many more parameters play a role in a real scenario. Also, not only properties of the fire are important, but also parameters of the environment,

such as type of soil, fertility of the soil and the distance of a zone to the nearest human settlement. A possible formula could then look like the following:

$$risk_{expected} = w_1 * FlameHeight_i + w_2 * \frac{1}{ArrivalTime_i} + w_3 * FireIntensity + w_4 * SoilValue.. \tag{2}$$

Here we have added FireIntensity and SoilValue. FireIntensity means the strength of the fire and SoilValue describes the value of the soil with regard to fertility. Of course, other parameters can be added.
Finally, we decided to use two different machine learning alogrithms. One time we made the predictions using a neural network. The other time by means of a multi-output regressor. In the latter method we proceeded in two different ways. Finally, we trained both models and measured and compared their performance using the following formula:

$$Error_{model} = \sum_{i=1}^{j} |risk_{model} - risk_{expected}| \tag{3}$$

The algorithm that has the smaller deviation performs better. We then use this algorithm to determine the risk of a zone with the Firefly algorithm. This way, we have determined a risk ratio under consideration of real data and not estimated, as it was originally done in the Firefly algorithm. By doing this, we hope to improve the overall performance of the algorithm. This will be described in more detail in later sections.

6.1 Neural Network

Neural networks have gained a lot of popularity in the last years. They are well suited for problems that are more complex with multiple variables and input data. This is exactly what we have. Therefore, we have also chosen to use these methods.
We use a simple model for this, as shown in the following graphic:

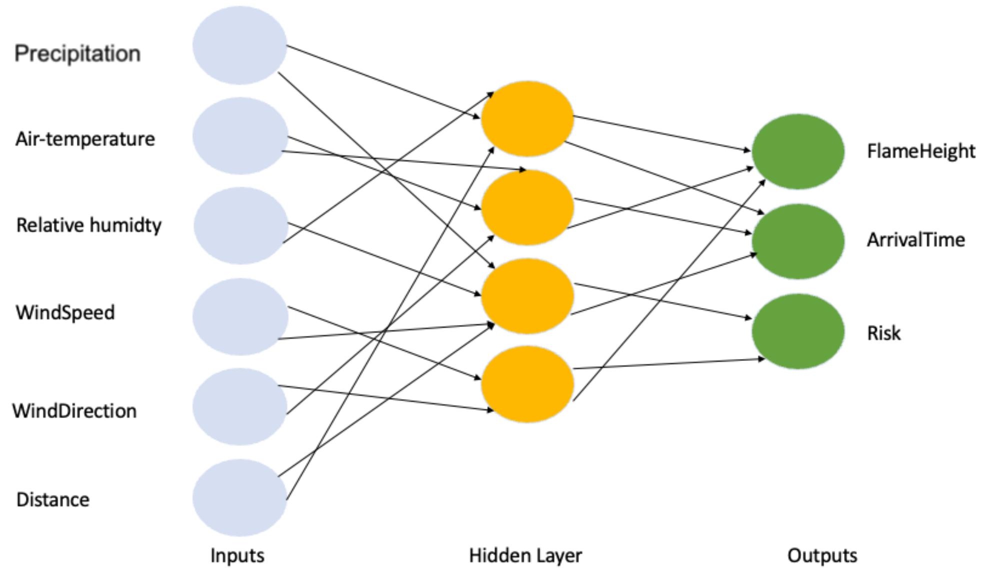

Fig. 3. Architecture of the neural network

This is a simple neural network, consists of inputs, outputs and a hidden layer. The following inputs are important here:

- Precipitation
- Air-temperature
- Relative humidty
- WindSpeed
- WindDirection
- Distance

These inputs have had the most influence in the coefficient analysis. Our output is three-dimensional. Based on the inputs, we would like to predict the following outputs:

- FlameHeight
- ArrivalTime
- Risk

We trained the neural network over n=500 steps. The expected loss can be seen in the following graph:

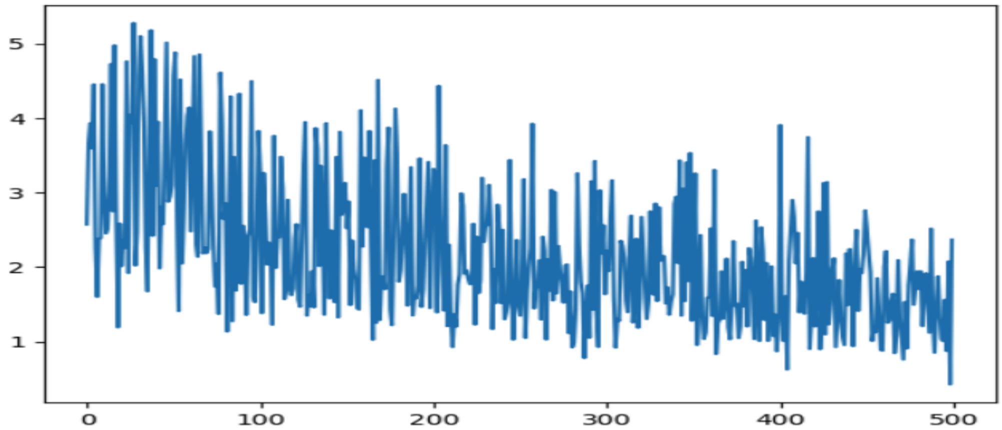

Fig. 4. Expected loss over n=500 steps

Despite the strong fluctuations, you can see that the expected loss decreases on average. This is good, because we can see a training success of the neural network. In total we have trained the neural network twenty times. The results are shown in the graph below:

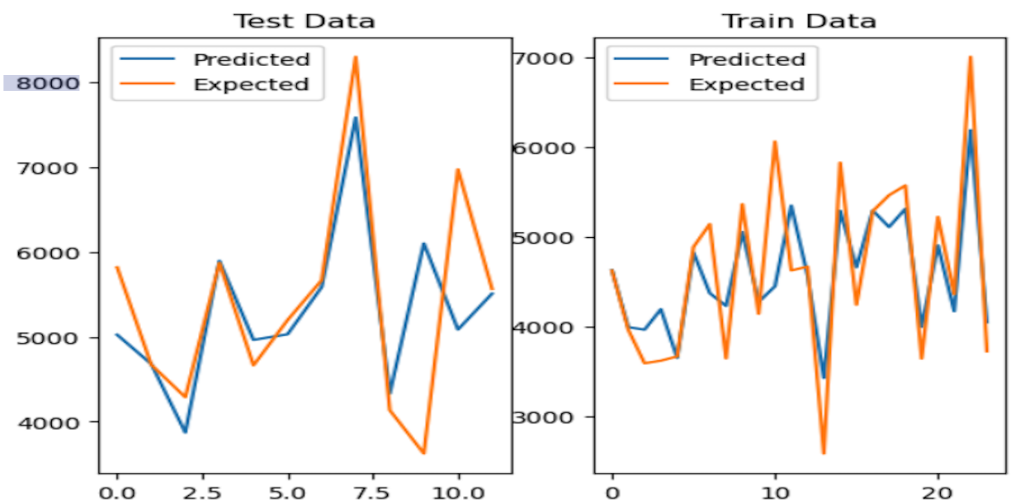

Fig. 5. Results of the neural network

The average error was 565. The left graph shows the result of the test data. The right graph shows the result of the training data. In each graph there are two graphs. The orange graph shows the expected result, which we know from our simulation. The blue graph represents the estimated result of the neural network. You can see well in both graphs that the

blue line follows the trends of the orange line.
The desired deviation between the blue and orange line is as small as possible. The smaller the deviation, the better the neural network performs. Because the deviations have a big influence on the real world. If the estimated graph runs over the real one, it means that the estimated risk is higher than it is in reality. As a result, one zone is given too much importance and then this zone is fought first, although another zone would be more important in reality. If the estimated graph is below the actual graph, a large risk is wrongly classified as a smaller risk. Thus, a large risk may be given a lower priority than a small risk.

You can see that the result of the training data tends to perform better than the test data. This is also due to the fact that the data set we use for training is larger. Nevertheless, the result is promising. In future work we hope to have larger data set so that the result will be further improved.

6.2 Multi-Output-Regressor

As a second method, we decided to use a multi-output regressor. The Multi-Output Regressor algorithm is a machine learning method based on supervised learning that is used to model the relationship between input variables and multiple output variables. The algorithm uses a RandomForest regressor, which is an ensemble learning method based on decision trees.
In detail, the algorithm works as follows: First, the RandomForestRegressor is trained with the input data to make a prediction for a single output variable. Then, this process is repeated for each of the output variables. The prediction for each of the output variables is then merged to produce the final output of the multi-output regressor.

In our case, the individual steps are as follows:

- Problem 1: Given X, predict FlameHeight
- Problem 2. Given X and FlameHeight, predict ArrivalTime
- Problem 3: Given X, Flame Height and ArrivalTime, predict Risk

With the explained algorithm, the following results came out:

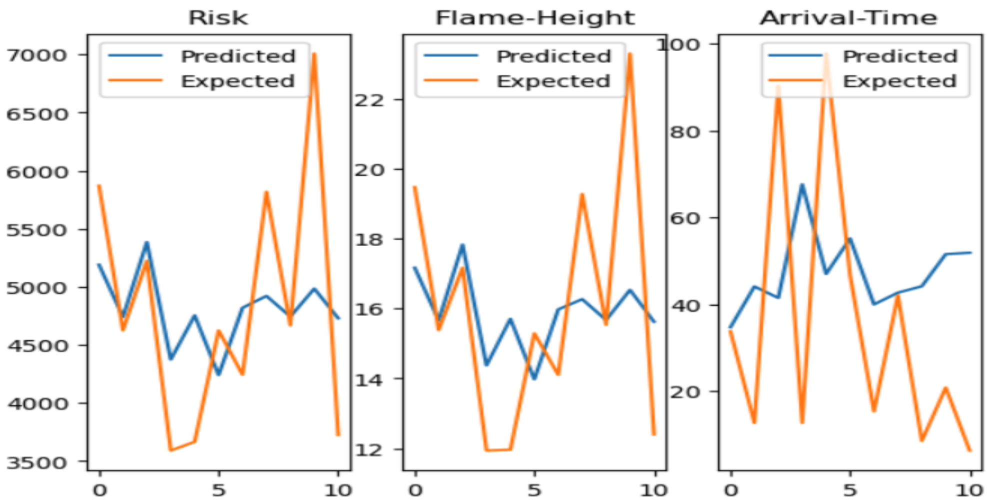

Fig. 6. Results of the Multi-Output Regressor

Here you can see three different graphics. The first graph shows the result of the parameter Risk, the second the result of the FlameHeight and finally the result of the ArrivalTime. You can already see with the naked eye that the distances of the graphs of expected and estimated are clearly further apart and that for each graph.
It is noticeable that underfitting usually occurs with the graphs. This may be due to various reasons. It is possible that we have too few features in our model. This can be improved in the future by adding more parameters to the equation. Also, as with neural networks, too little training data can be a reason for poorer performance.

The ArrivalTime graph can be interpreted as follows: You can see that the estimated graph is mostly below the real graph. This means that the ArrivalTime is estimated shorter than it actually is. This gives a higher priority to zones where the fire may not have arrived yet.

The plot of the FlameHeight also shows deviations in both directions. If the estimated graph runs above the expected graph, the FlameHeight is estimated too high and a zone is assigned too high importance. If the estimated graph runs below the expected graph, the FlameHeight is estimated too low and thus a zone is assigned too low an importance.

The interpreation of the risk is the same as for the neural network.
According to the formula, the average error here was n=707.

6.3 Firefly Algorithm

6.4 Steps

First Firefly formulates allocation of drones as a maximum set coverage problem and uses a greedy algorithm to allocate.

In the second step it observes the potential damage and the utility of the measured zones. Third, Firefly uses estimated and calculated values to solve firefighter allocation as knapsack problem.

6.5 Performance Measurement

In order to compare the performance of the Firefly algorithm, we had to come up with a suitable method. Ultimately, the goal of the algorithm is to keep the damage as low as possible. This is achieved by sending enough firefighters to a zone so that the fire is extinguished. To decide which zone to choose, the reward function plays a crucial role. This is determined as follows:

$$\hat{k_{i,t}} = k_{it} \tag{4}$$

$$\hat{k_{i,t}} = \frac{1}{t-1} * K_{i,t-1} \tag{5}$$

As you can see, there are two cases. In the first case, a zone is detected by a drone. Thus, the estimated reward is equal to the real one, since it is determined by the drone. If it is not possible for a drone to capture a zone, the reward is estimated by the second equation. In reality, the second case will probably occur very often, since fire areas are often very large and there are only a limited number of drones. Also, one has to consider that drones have to be loaded and are not steadily available. Therefore, the second case is likely to occur more often. Exactly this one, is only estimated here. Here, the cumulative reward, from a point in time earlier is reverted to and this is assigned to the zone. Exactly here we hope to be able to improve the algorithm by our prediction of the estimated reward. Reward here is equal to our risk. Because of the huge scale of forest fires and their damages, small improvements can have a big impact. To compare the success of both methods, we have considered the following:

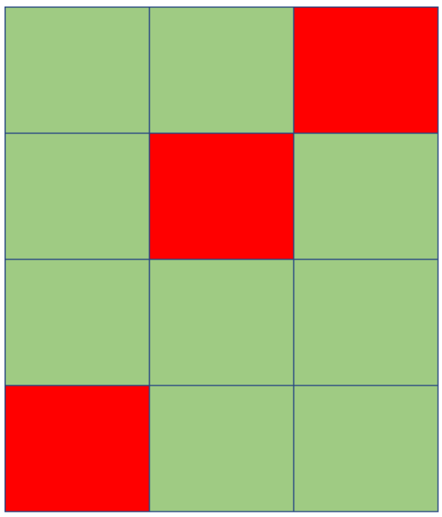

Fig. 7. red: high-risk area; green: save area

In the figure above you can see an area that has been affected by a fire. This area consists of twelve individual zones. Red means that this zone is currently burning and has a high priority. Green means a zone is safe at the current time. Thus we have x=3 red zones and y=9 green zones. Now our assumption is that we have exactly as many emergency forces available to get exactly the x=3 zones under control. We assume that five firefighters are needed to extinguish one fire. Consequently, we have 15 forces at our disposal. An optimal result would be the following: The Firefly algorithm correctly identifies all red zones, meaning it gives them the highest risk. Consequently, five firefighters are sent to a red zone to fight it successfully. To compare the success of both methods, it would be good to have some kind of factor. We came up with the following idea:

$$success_{method} = \frac{z}{x} * 100 \tag{6}$$

X is the number of red zones as already mentioned. Z is the number of red zones correctly identified by a method. Thus we get a factor that we can finally compare.
In the optimal case, the result of the risk assessment would be as follows:

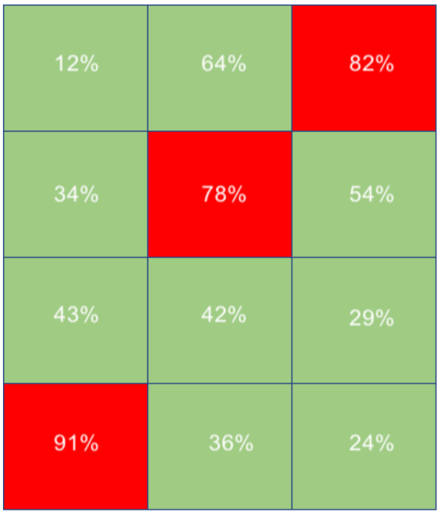

Fig. 8. red: high-risk area; green: save area

The red zones have been correctly classified as the most important and the task forces are assigned to them.

7 Conclusion and Future Work

Our project had many exciting phases. It started with the challenges of collecting high quality data. It was important to collect good data, because it plays a big role in later steps. To collect data, we familiarized ourselves with the tool Spark. Here we simulated a wildfire. We did this process several times. Each time we changed the start parameters. So we got different outpus and a good dataset to work with. The next step was to analyze the collected data. For this we used the software Weka. Here it was important to determine the correlations between input and output parameters. In the next step we tried to determine the risk used for the Firefly algorithm. For this we chose two different machine learning algorithms. These were the classical neural networks and a multi-output regressor. Both methods worked well. Nevertheless, there is still some room for improvement. A better result deliver here the neural network. This had an error of n=565. In contrast, the latter algorithm had an error of 707. For this reason, we use the neural network to determine the risk of a zone, which is then used in the Firefly algorithm. The Firefly algorithm deals with the wildfire problem. More precisely, it is a method to combat it. It also takes into account the changing environment and uncertainty about the fire. It first formulates the drone allocation as maximal set coverage problem, and uses a greedy algorithm to allocate the drones. Then by using the observed and estimated data Firefly solves the firefighter allocation problem as a knapsack. For the latter, the reward of a zone plays a crucial role. This is tried to maximize. This is exactly determined by equation (4). If this is not possible, it is estimated only by the cumulative reward equation (5). This is exactly where we have proposed an improvement through machine learning. This estimation is based on real simulated data and already works well. To compare the performance of both methods, we came up with equation (6). In future work, it is important to collect more data. It was seen in the results of the machine learning algorithms that there was often underfitting or overfitting. We suspect the skin reason for this was a data set that was too small. The more data we collect in the future, the better our results will be. We expect to achieve much more accurate results with more data. We also want to add more machine learning algorithms in further work. We will then compare these with the two we already have. In future work we will compare the performance of the two variants of reward determination.

8 References

Environmental circumstances surrounding bushfire fatalities in Australia 1901-2011

Probabilistic allocation and scheduling of multiple resources for emergency operations; a Victorian bushfire case study

Resource Sharing for Control of Wildland Fires

Fighting Wildfires under Uncertainty: A Sequential Resource Allocation Approach

Using Social Media to Detect and Locate Wildfires

An Integer Programming Model to Optimize Resource Allocation for Wildfire Containment

Fire intensity, fire severity and burn severity: A brief review and suggested usage

8.1 Links

Spark Documentation

CSIRO research

Spark content

Weka wiki

Weka

WWF

UNEP Report

References