Problem Chosen	2022 MCM/ICM Summary Sheet	Team Control Number
D		00000000

Summary

Australia is undergoing huge wildfires in every state. To protect people and safety and property, we establish a model to use two types of drones to help Country Fire Authority (CFA) conduct "Rapid Bushfire Response", and front-line personnel communicate with Emergency Operations Center (EOC).

We use Victoria Fire Report Data[2], using fire report places in certain time period to represent locations where fires happened on average. This allows us to take into account of factors such as fire size and frequency, economic cost and safety, weighted region area covered by drones.

We set Fast Response Model for deployment of drones for fast response, we quantify the coverage with a weighted quantity. We build a model to investigate its' relationship with the economic cost. For a certain coverage by drone, we devise a strategy to distribute SSAs by using k-means with special distance function and we use minimum spanning tree to distribute repeaters to connect them. We show the mathematical properties to such distance to ensure the correctness of our algorithm.

We set Fire Prediction Model for second part of problem. we divide Victoria into several zones, and use statistics of zones in different time to form a time series. We predict the time series using convolutional Long Short-Term Memory (convLSTM).

We set Pearl Model and Spur Model for Deployment of drones for front-line personnel in different circumstances, we use separate deployment strategies for small and big sized fire considering the effect of terrain.

The sensitivity analysis shows robustness in our model. Meanwhile, we combine all the models to finish the annotated Budget Request to help CFA with acceptable cost.

Keywords: Clustering, Minimum Spanning Tree, ConvLSTM, Terrian;

Contents

1 Introduction **3**

1.1 Background 3

1.2 Problem Restatement 3

1.2.1 Limits 3

1.2.2 Targets 3

1.3 Our Work 4

2 List Of Symbols **4**

3 General Assumptions **5**

4 The Models **6**

4.1 Fast Response Model 6

4.1.1 Data Pre-processing 6

4.1.2 Deploying SSA 7

4.1.3 Deploy Repeaters 10

4.2 Fire Prediction Model 13

4.2.1 Data Pre-processing 13

4.2.2 Build Map with Fire Index 13

4.2.3 Time series construction 14

4.2.4 ConvLSTM 15

4.2.5 Model Fitting 17

4.3 Pearl and Spur Model 17

4.3.1 Pearl Model 17

4.3.2 Drones' locations strategy explanation 18

5 Conclusions **21**

6 Model Evaluation And Improvement **21**

6.1 Strength 21

6.2 Weakness 21

6.3 Improvement 21

1 Introduction

1.1 Background

Wildfire spreads rapidly in Australia. In fire season, it's devastating for people's safety and properties. Victoria's Country Fire Authority (CFA) uses different means to protect its people. Drones carrying high definition & thermal imaging cameras and telemetry sensors were sent for surveillance and situational awareness (SSA). Drone repeaters, transceivers that automatically rebroadcast signals at higher powers can help connect Emergency Operations Center (EOC) with SSA and front-line employees with VHF/UHF bands.

1.2 Problem Restatement

1.2.1 Limits

- Drone-related
  - Cost $1000 per drone
  - Flight Range 30 km
  - Transmission Range 20 km
  - Flight Speed 20 m/s
- Fire-related
  - Size
  - Frequency

1.2.2 Targets

- Deployment of drones for fast response
  - Reduce cost
  - Increase weighted coverage

- Fire Prediction
- Deployment of drones for front-line personnel in different circumstances
  - Build models for different fire size
  - Build models considering different terrians

1.3 Our Work

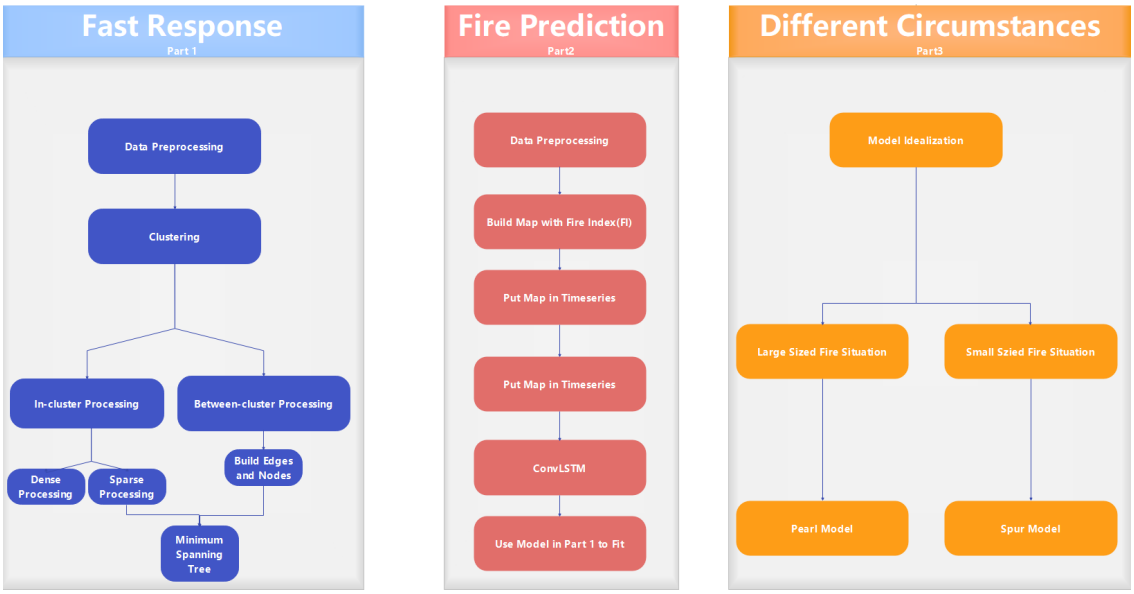

Figure 1: Our Work

2 List Of Symbols

Definition	Description
$dist$	Distance between two points in Euclidean coordinate system
R_e	Radius of earth
$\Delta\varphi_{lat}$	Thange of latitude
$\Delta\lambda_{lon}$	Thange of longitude
eps	Tf the distance between two points is lower or equal to (eps), these points are consic
$minPoints$	The minimum number of points to form a dense region.
x_0	Distance to drone when the fire is recorded in our data

Definition	Description
x_1	Distance to drone when drone detect fire
$dis_{j,t}$	The shortest distance of fire location indexed j to the nearest drone at distance x fro
r_c	The radius of drone in idealized condition.
$vs_{j,x}$	The spread rate of fire at the rim of fire area indexed j at distance x from the drone
$p_{j,x}$	The probability for drones to detect rim of fire location j at distance x from the dron
v	Stable spread rate which is assumed to simply the model
WCL	Weighted coverage loss : describing the loss the weighted area for a deployment str
r_o	Outer-radius of drone, meaning the furthest distance the drone can detect, that is, 5
$tWCL$	Threshold for WCL to determine how much SSA should be deployed.

3 General Assumptions

- We use one year data in Victoria with data provided by Earth Data to represent the general cases in Australia. However, our model to this case adapt to arbitrary cases, so it's without losing generality.
- We assume once the fire is within the detective range of drones, it will be found out without delay.
- We assume the spread rate of fire is stable and at a certain value, which is not the case in real world, but one can use the original formulae given in the model to simply modify the model.
- Only 20 years of data is used for machine learning, the error produced is within the acceptable range.
- The terrain situation can be more complicated in real world, we idealize mountain and other barriers as parabolic-like object.

4 The Models

4.1 Fast Response Model

To discuss the possible deployment of drones in order to detect fire and transmit the signal to EOC, we design Fast Response Model to maximize coverage and minimize the cost. To represent the fire distribution, fire frequency and fire size, we come up with several well-designed indices and use fire location in certain period to represent those factors with minimum lost of information. Since it's not economically efficient to cover all the land of Victoria because the drones are able to move and the fact that fire can spread and then be detected, we use weighted covering lost(WCL) to represent the cost for not covering all the possible locations of fire. We use the data in 2020 for case study, but the strategy we adapt and the data we compute is generic and can be used in various situation.

After sensitivity test, we proved the robustness of the model. It can be showed that the Fast Response Model can be used in different size of fire, different frequency of fire, and different distribution of fire in state of Victoria and other places in the world.

4.1.1 Data Pre-processing

For the sake of CFA, our model should only be considering the fire situation within the range of state of Victoria. The data we obtained from NASA database is contains noise and locations out of border. The first step of data pre-processing is meant to sift out all the illegal point with criteria mentioned above. Considering the spatial location of noise point, we use DBSCAN clustering with ball tree [8] algorithm, and is implemented by sci-learn project[7]. Since the data contains latitude and longitude, to define the distance function for clustering one need to use the haversine formula[1] to calculate the great-circle distance between two points.

$$dist_{i,j} = 2 \cdot R_e \cdot \arctan\left(\sqrt{\frac{\sin^2(\frac{\Delta\varphi_{lat}}{2}) + \cos\varphi_i \cdot \cos\varphi_j \cdot \sin^2(\frac{\Delta\lambda_{lon}}{2})}{1 - (\sin^2(\frac{\Delta\varphi_{lat}}{2}) + \cos\varphi_i \cdot \cos\varphi_j \cdot \sin^2(\frac{\Delta\lambda_{lon}}{2}))}}\right) \quad (1)$$

This ensures the correctness of clustering.

To define a noise point which is inefficient to cover it, we define two variables eps and $minPoints$ according to DBSCAN conventions.

To more easily obtain the optimized value, we first normalize data with standard normalization, then we set

$$\begin{cases} eps = 0.15 \\ minPoints = 8 \end{cases} \tag{2}$$

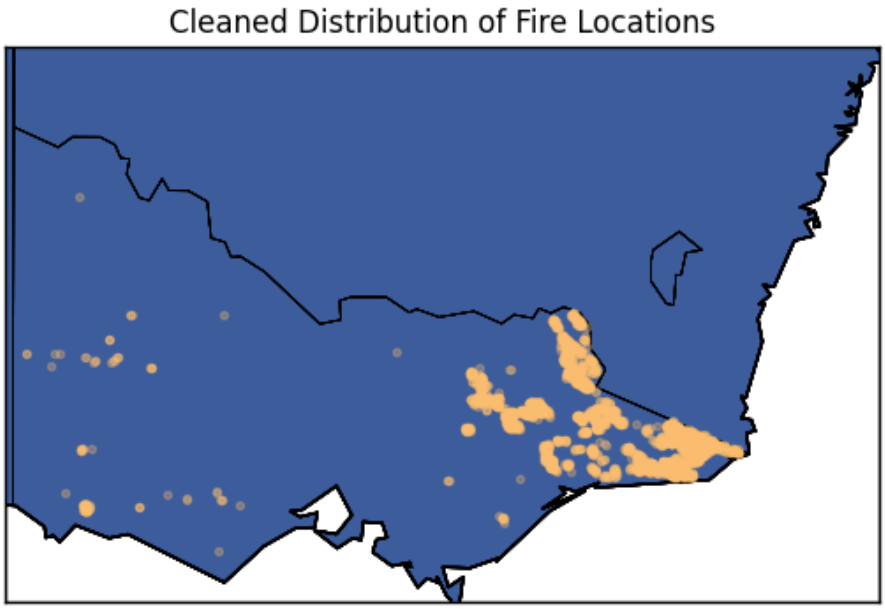

Figure 2: Cleaned Fired Distribution

4.1.2 Deploying SSA

To deploy drones in a way that reaches the target of fast response, we first need to quantify the target using one index, which we define it as weighted covering lost(WCL).

$$WCL = \sum_j \left(\int_{x0}^{x1} (x - r_c) \cdot (1 - p_{j,x}) \cdot vs_{j,x} \right) dx \tag{3}$$

To simplify our model, we assume $vs_{j,x} = v$, which is a stable value, then we have.

$$WCL = \sum_j \left(\int_{x0}^{x1} (x - r_c) \cdot p_{j,x} \cdot v \, dx \right)^2 \tag{4}$$

In order to simply the model as well as simulate the distribution of p_j, we use Ridge Distribution to set $p_{j,x}$, which is the probability of rim of fire at the position which is x km from the nearest SSA, as following

$$p_{j,x} = \begin{cases} \frac{1}{2} - \frac{1}{2} \sin \frac{\pi}{r_o}(x - \frac{r_o}{2}), & 0 \leqslant x \leqslant r_0 \\ 0, & x > r_0 \end{cases} \tag{5}$$

This gives us

$$WCL = \sum_j \left(\int_{x0}^{x1} (x - r_c) \cdot (\frac{1}{2} + \frac{1}{2} \sin \frac{\pi}{r_o}(x - \frac{r_o}{2})) \cdot v \, dx \right)^2 \tag{6}$$

We use the concept of substitution distance($sdist$) to investigate the deployment strategy.

$$sdist = \left\| \int_{x0}^{x1} (x - r_c) \cdot (\frac{1}{2} + \frac{1}{2} \sin \frac{\pi}{r_o}(x - \frac{r_o}{2})) \cdot v \, dx \right\| \tag{7}$$

We define $sdist$ in a way that guarantees $sdist$ is positively correlated to x which is the distance to the center, that is, the place where the nearest drone is deployed.

To balance economical costs and safety, we set a threshold for WCL, $tWCL$ which is currently set to a certain value in our later investigation, but it can be adjusted according to real situation. It will be illustrated more thoroughly in the following section about sensitivity and robustness. We use modified $k - means$ cluster to determine the positions of SSAs, which is described as follows

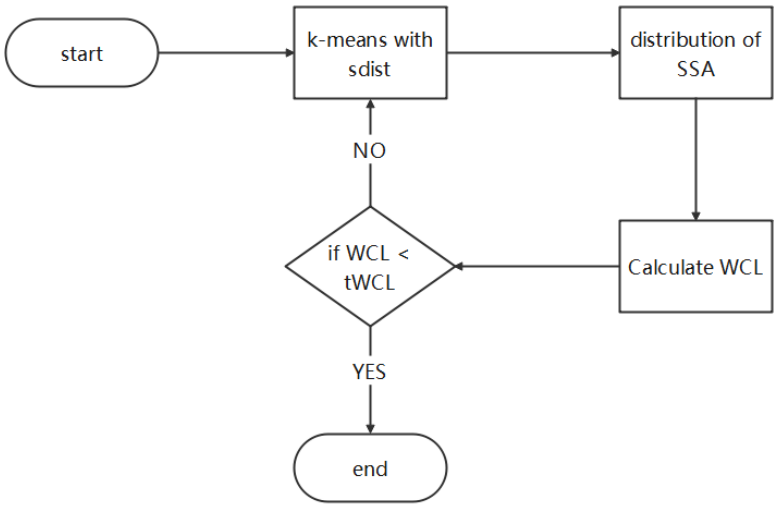

Figure 3: deploy SSA flowchart

This allows us to cluster the locations to their respective drones. k-means algorithm is used here since the $sdist$ is positively correlates to distance. So the correctness of the algorithm can be ensured.

(a) clustering of fire location (b) SSA distribution

Figure 4: SSA and fire locations

Given the fire location distribution, we plot the SSA's location as above. The range is marked as well. It can be observed that the fire is frequent and in large scale at the east of state of Victoria. Our distribution perfectly fit the situation can reduce WCL to acceptable level.

4.1.3 Deploy Repeaters

The second part of Fast Response Model is connecting all the SSAs using the least repeaters. We solve this problem by divide and conquer and with the help of minimum spanning tree.

For Dense Cluster, we choose to shrink several SSAs into one, since when it's dense, one repeater can cover multiple SSAs. Then we build minimum spanning tree on the repeaters.

For Sparse Cluster, we choose to build minimum spanning tree directly.

Combined both situation, we succeed in connecting all SSAs in a way that guarantees both the stability of connection and minimum economic cost.

Dense Cluster For dense cluster, we choose shrink point strategy. For every dense cluster which is formed by a set of SSAs, we use an algorithm to settle the distribution of core repeater.

A core repeater is defined as, the repeater that can receive signal from SSAs.

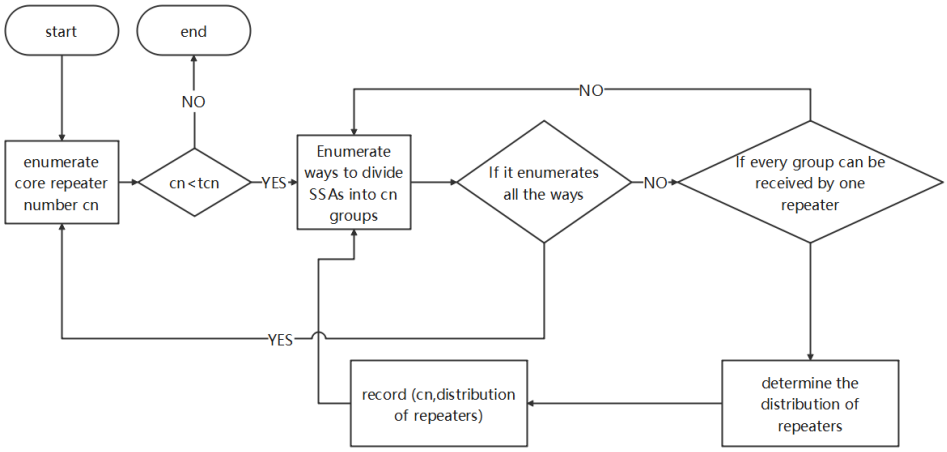

Figure 5: Strategy for Dense Clutser

We want the combination of core repeaters to be in a way that allows the minimum number of repeaters to be distributed and all the SSAs' signal can be received. There seems to be no elegant solution to this problem because it is NP

problem. However, the small number of SSAs allows us to use brute force to compute such distribution.

We define cn to be the number of core repeaters needed, $nSSA$ to be the SSA that are distributed, we define tcn to be the maximum number of core repeaters that are allowed, it's obvious that $tcn \leqslant nSSA$. For simplicity of implementation, we set $tcn = nSSA$

Sparse Cluster and Final Linking Now there are only sparse SSAs and sparse repeaters, it's natural to link them by Kruskal algorithm to make a minimum spanning tree.

The first step is to build the graph with nodes and edges.

- Nodes are SSAs and repeaters, which is reasonable since they are sparse and can be discretized.
- Edges have weight defined below. Considering if the range of drones are tangent to each other, the transmission stability will be lowered, we introduce transmission random factor trf

$$Ew_e = \max\left(\left\lceil \frac{dis_{i,j} - 2r_c - trf_e}{2r_c} \right\rceil, 0\right) \tag{8}$$

Transmission factor is defined below to reduce the situation where ranges of drones are tangent, where $randi(x)$ function is to randomly produce a real number within interval $[0, x]$

$$trf_e = randi\left(1 - \left(\left\lceil \frac{dis_{i,j} - 2r_c - trf_e}{2r_c} \right\rceil - \frac{dis_{i,j} - 2r_c - trf_e}{2r_c}\right)\right) \tag{9}$$

Then we use Kruskal Algorithm[6] as described below to make a spanning tree.

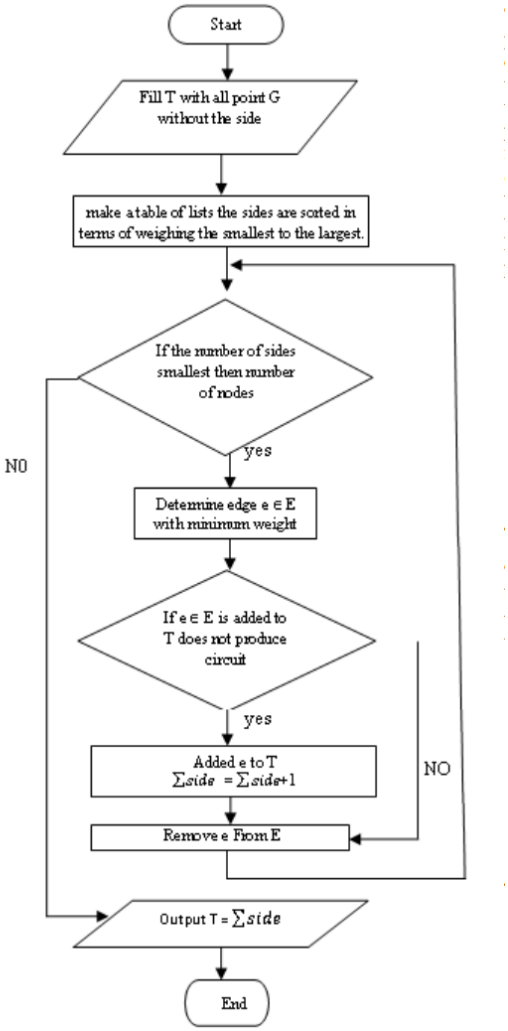

Figure 6: Strategy for Dense Clutser

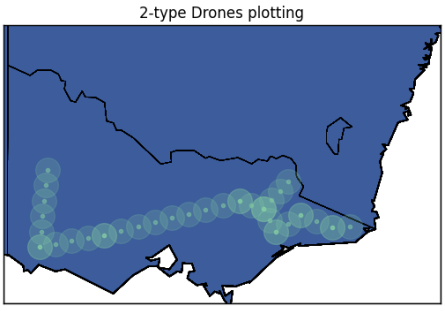

Figure 7: The general drone distribution

Because of the introduction of trf_e and the use of divide and conquer strategy, the outcome is not only stable but also efficient as shown.

4.2 Fire Prediction Model

4.2.1 Data Pre-processing

The data source used in this task is from Moderate-resolution Imaging Spectroradiometer(MODIS) provided by NASA. The obtained time series data of Australia wild-fire from 2003 to 2020 is saved in CSV format. The task is to drop all data with confidence less than 80, and divide them monthly. A high threshold for confidence is adopted to reduce the noise of input data and make the prediction result more reliable. Given the intensity of wildfires, it makes sense to combine data from the same month to create heat maps. All mentioned operations are based on Pandas in Python.
The data source used in this task is from Moderate-resolution Imaging Spectroradiometer(MODIS) provided by NASA. The obtained time series data of Australia wild-fire from 2003 to 2020 is saved in CSV format. The task is to drop all data with confidence less than 80, and divide them monthly. A high threshold for confidence is adopted to reduce the noise of input data and make the prediction result more reliable. There are two main reasons for selecting monthly divided data:

1. Monthly divided data has a long time span, which can provide a long enough time series.
2. Monthly divided data has uniform time interval, which is convenient for statistical modeling of time series.

All mentioned operations are based on Python's framework Pandas.

4.2.2 Build Map with Fire Index

Australian Bureau of Statistics offers digital boundary files of all states. By reading the shape file and monthly wild-fire data in MATLAB R2021b, it's easy to use filterm function to drop all data points out of the state Victoria, and map all points to a 109x185 matrix, where 20 terms in each dimension corresponding to one degree in geography.

The Heatmap function can build maps in an intuitive and easily machine-learned form. By building maps for every month in 17 years, 204 maps are obtained. The figure below shows the wild fire in Victoria in 2003.

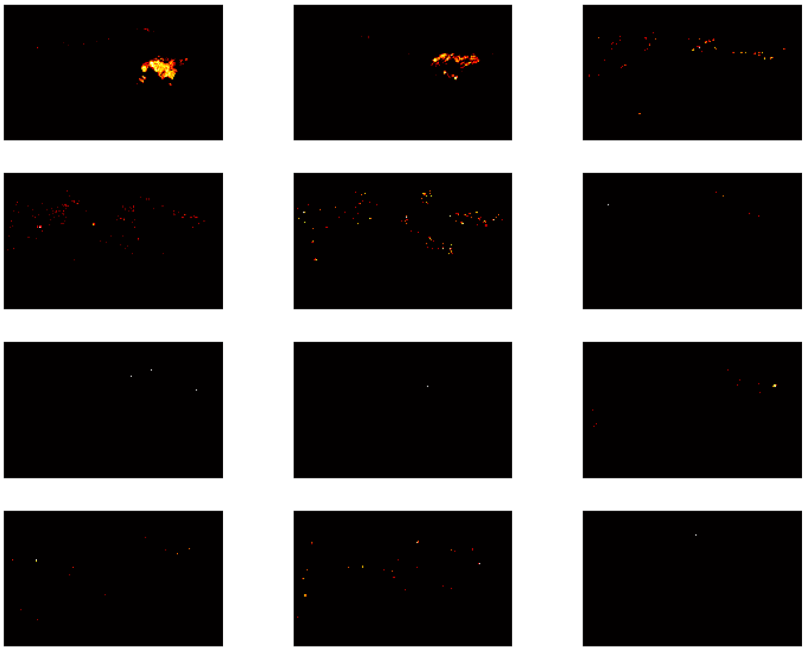

Figure 8: good

The RGB channel values in the picture are given by the following formula:

$$R_{x,y} = \begin{cases} \left\lfloor 255 \frac{\sqrt[4]{heat_{x,y}}^3}{\max(\sqrt[4]{heat})} \right\rfloor & \sqrt[4]{heat_{x,y}} < \frac{1}{3}\max(\sqrt[4]{heat}) \\ 255 & \sqrt[4]{heat_{x,y}} \geqslant \frac{1}{3}\max(\sqrt[4]{heat}) \end{cases} \tag{10}$$

$$G_{x,y} = \begin{cases} 0 & \sqrt[4]{heat_{x,y}} \leqslant \frac{1}{3}\max(\sqrt[4]{heat}) \\ \left\lfloor 255\left(\frac{\sqrt[4]{heat_{x,y}}^3}{\max(\sqrt[4]{heat})} - 1\right) \right\rfloor & \frac{1}{3}\max(\sqrt[4]{heat}) < \sqrt[4]{heat_{x,y}} < \frac{2}{3}\max(\sqrt[4]{heat}) \\ 255 & \sqrt[4]{heat_{x,y}} \geqslant \frac{2}{3}\max(\sqrt[4]{heat}) \end{cases} \tag{11}$$

$$B_{x,y} = \begin{cases} 0 & \sqrt[4]{heat_{x,y}} < \frac{2}{3}\max(\sqrt[4]{heat}) \\ \left\lfloor 255\left(\frac{\sqrt[4]{heat_{x,y}}^3}{\max(\sqrt[4]{heat})} - 2\right) \right\rfloor & \sqrt[4]{heat_{x,y}} \geqslant \frac{2}{3}\max(\sqrt[4]{heat}) \end{cases} \tag{12}$$

4.2.3 Time series construction

4.2.4 ConvLSTM

Time series data prediction refers to learning past time series and predicting future changes. Traditional Neural networks cannot solve the problem of time-axis variation, so RNN (Recurrent Neural network) is developed [5].

However, due to the poor performance of classical RNN in extracting long time series information and the limited time series information extracted, Hochreiter developed LSTM network model [4]. In classical RNN, gates structure is added to selectively add and delete the past timing information, and input gate, output gate and forgetting gate are added to control the input and output of data of this unit (an LSTM cell is a basic unit) and the increase and decrease of the output information of the previous unit respectively. The LSTM formula is expressed as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{X}_t + \mathbf{W}_{hi}\mathbf{H}_{t-1} + \mathbf{W}_{ci} \circ \mathbf{C}_{t-1} + b_i) \tag{13}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{X}_t + \mathbf{W}_{hf}\mathbf{H}_{t-1} + \mathbf{W}_{cf} \circ \mathbf{C}_{t-1} + b_f) \tag{14}$$

$$\mathbf{C}_t = \mathbf{f}_t \circ \mathbf{C}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}_{xc}\mathbf{X}_t + \mathbf{W}_{hc}\mathbf{H}_{t-1} + b_c) \tag{15}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{X}_t + \mathbf{W}_{ho}\mathbf{H}_{t-1} + \mathbf{W}_{co} \circ \mathbf{C}_{t-1} + b_o) \tag{16}$$

ConvLSTM is a variant of LSTM proposed on the basis of LSTM. It replaces the fully connected state between the input layer and the hidden layer and between the hidden layer and the hidden layer of LSTM with the convolution connection, which makes full use of the spatial information that LSTM cannot. LSTM needs to transform image data into one-dimensional vector when processing image data, and cannot process spatial structure information of original image data. Compared with LSTM model,Conv LSTM can better extract spatial and temporal structure information from time series images. ConvLSTM model formula is expressed as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi} * \mathbf{X}_t + \mathbf{W}_{hi} * \mathbf{H}_{t-1} + \mathbf{W}_{ci} \circ \mathbf{C}_{t-1} + b_i) \tag{17}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf} * \mathbf{X}_t + \mathbf{W}_{hf} * \mathbf{H}_{t-1} + \mathbf{W}_{cf} \circ \mathbf{C}_{t-1} + b_f) \tag{18}$$

$$\mathbf{C}_t = \mathbf{f}_t \circ \mathbf{C}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}_{xc} * \mathbf{X}_t + \mathbf{W}_{hc} * \mathbf{H}_{t-1} + b_c) \tag{19}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo} * \mathbf{X}_t + \mathbf{W}_{ho} * \mathbf{H}_{t-1} + \mathbf{W}_{co} \circ \mathbf{C}_{t-1} + b_o) \tag{20}$$

The symbol meaning in the formula is the same as that in LSTM. The full connection of input variables is replaced by convolution operation. According to the internal structure of ConvLSTM in figure, it can be seen that input gate, output gate and forgetting gate all carry out convolution operation for input and hidden layer.

$\mathbf{W}_{ci} \circ \mathbf{C}_{t-1}$, $\mathbf{W}_{cf} \circ \mathbf{C}_{t-1}$ and $\mathbf{W}_{co} \circ \mathbf{C}_{t-1}$ in the formula indicate that the input, output and forgetting gates are connected to the Peephole[3] of the previous cellular state. As shown in the figure, the Peephole connection adds cell state information to each gate. Since the unit may have a door state of 0, which results

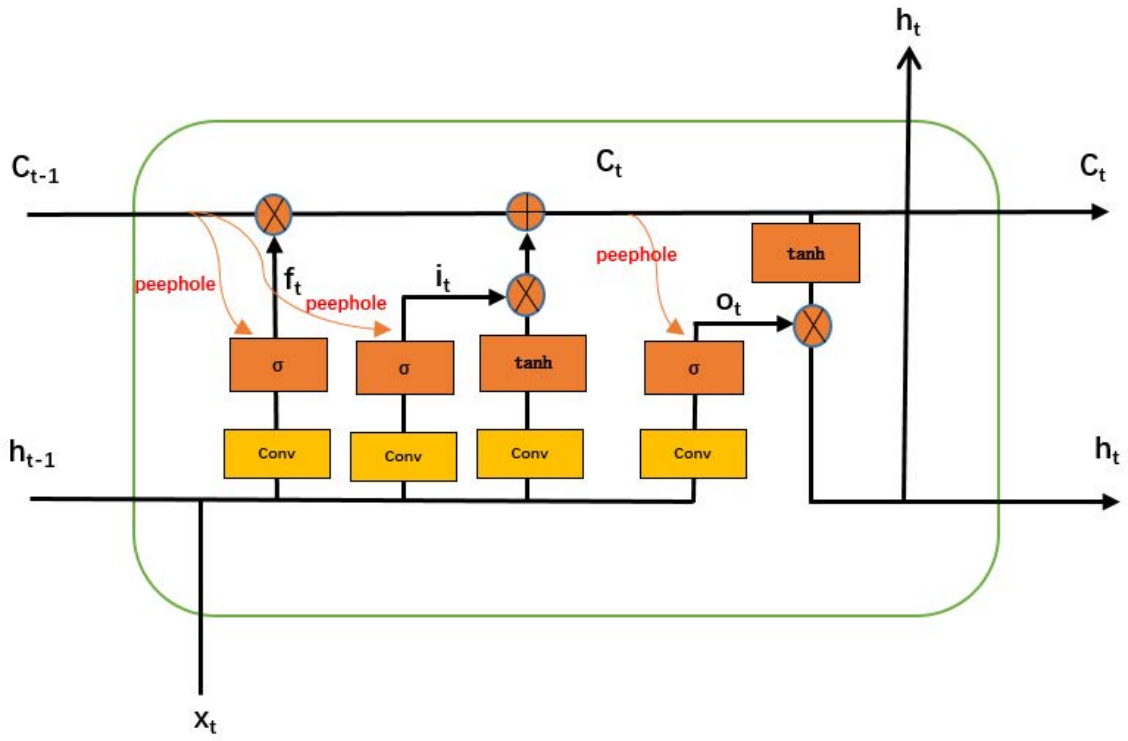

Figure 9

in a lack of important information, adding the Peephole operation can improve this shortcoming.

Based on the deep learning framework Pytorch, the ConvLSTM is constructed using Python language, and the experimental equipment environment is NVIDIA GeForce GTX1080 GPU.

4.2.5 Model Fitting

4.3 Pearl and Spur Model

4.3.1 Pearl Model

2019, Victoria, Australia suffered a severe bushfire. On 1th January, houses were burned to the groung as NSW fire spread to Corryong in the northern Victoria. Corryong, the town which is surrounded by Mount Mitta and Wabba Wilderness Park was in great in danger.

In order to determine our model for optimizing the locations of hovering VHF/UHF radio-repeater drones for fires of different sizes on different terrains, we need to consider various terrains, including hills, plain and mountains. Corryong, the city lying in the basin, is perfect for our optimization. Therefore, we will take Corryong for an ecxample to explain our location strategy.

We assume the fire is happening in the area surrounded by the red circle, the area is much larger than the drones hovering range. Our basic strategy is to supervise the edge of the bushfire area. Therefore , we will need the "Boots-on-the-ground" Forward Teams be at the front lines of the fire events carrying the VHF/UHF. Conisdering various situations, there is always no definitely perfect strategy to guide the teams to distribute. Thus, we consider all random situations for the distribution of the firefighters.

Based on the bushfire area, we can get a latitude function along the periphery of the enclosed area, which is shown as below.

This function has a x-axis which represents the distance along the periphery of the area from certain point, and a y-axis which represents the latitudes of the point. Considering the effect of latitude is significant to the height of drones should be, so that they avoid the signal loss caused by terrains as possible as

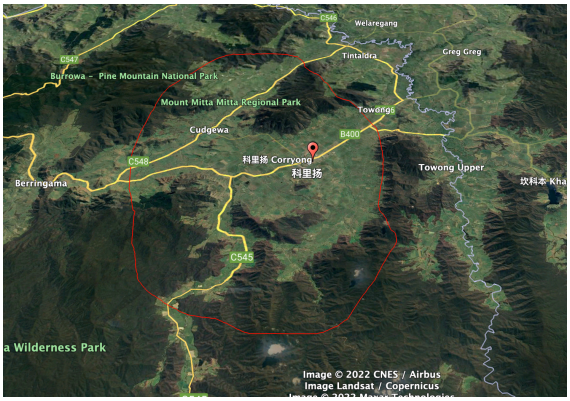

Figure 10: Hypothetical bush fire area happening in the Corryong

Figure 11: latitude along the peripher

they can.

We choose to use the Monte-Carlo algorithm to analyze the effect of firefighters' distribution to the locations of the hovering drones. For the first step, we will distribute some fire fighter at the front line of the bushfire area for simulation. Below is one condition considered.

Green points represent fire fighters carrying VHF/UHF (9 firefighters in simulation)

From the results provided by the Google earth pro, we obtain the function of the distance along the periphery and the latitude

$$H(x_i) \tag{21}$$

4.3.2 Drones' locations strategy explanation

There are some basic principles we need to follow to settle the drones

1. The distance of two drones can't be over 20 km, which is maximum range

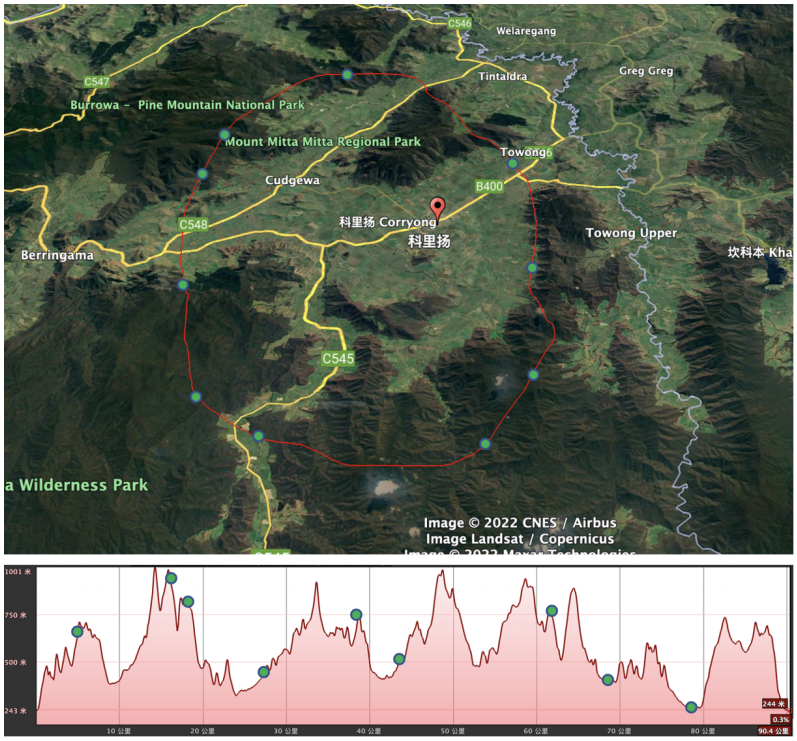

Figure 12: Distribute fire fighters ramdomly

that transceivers can spread and receive.

2. Every individual fire fighter must be received by at least one drone so that they can keep connected.

3. Drones should be settled along the periphery of the bushfire area.

4. Drones should be in the reasonable height so that the radio signal won't be interrupted by the terrain obstacles, for example hills between.

Therefore we have the drones' settling strattegy as below

$$sdist = \tag{22}$$

x_i means the location along the periphery; d_i represents the distance required along the periphery that makes the point x_{i-1} moving forward for 20km in the actual distance; sigma is the factor which make drones get closer to the previous one due to the terrain.

After these analysis, we can get the distribution of the drones:

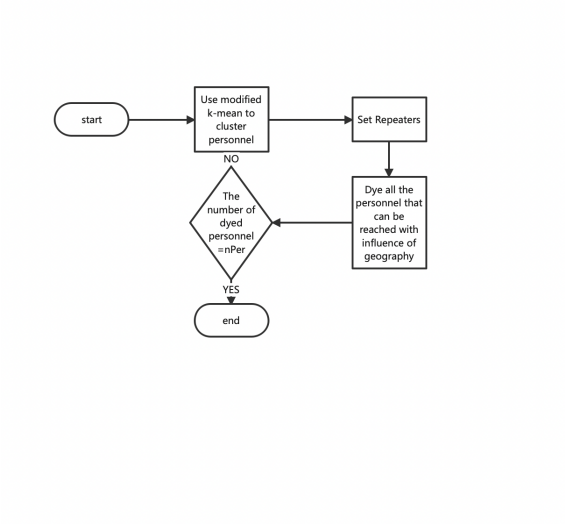

Figure 13: mindmap of drones strategy

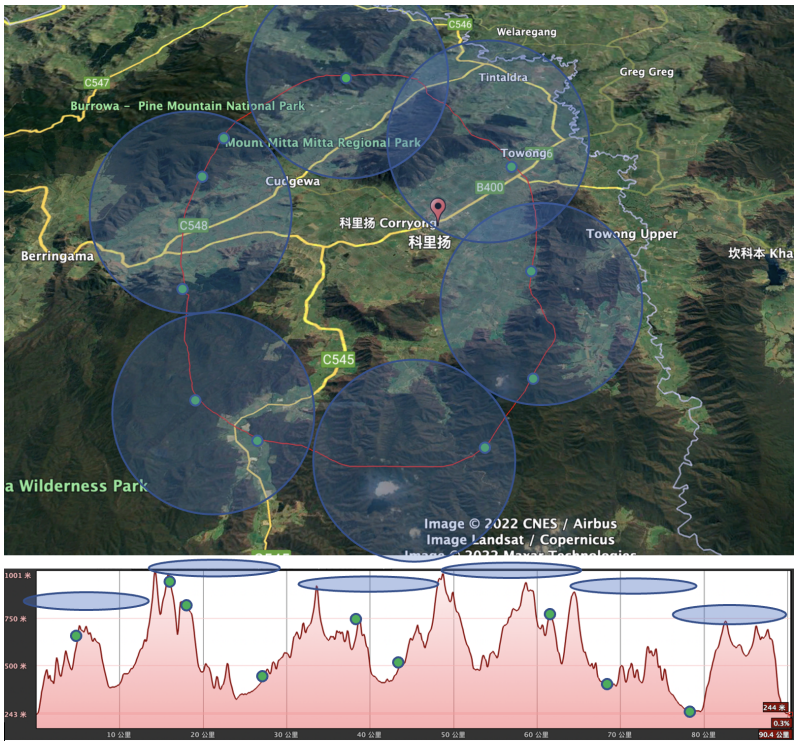

Figure 14: Simulated distribution of drones
Blue circles show the radio range of the transceivers on the drones

Blue circles and lines show the radio range of the transceivers on the drones

Drones' location strategy common derivation

Above is a specific situation of the distribution of the drones, in more common situations, the distribution of drones is related to the terrain and distribution

of the fire fighters.

5 Conclusions

Conclusions

6 Model Evaluation And Improvement

6.1 Strength

6.2 Weakness

6.3 Improvement

References

[1] Calculate distance, bearing and more between latitude/longitude points. http://www.movable-type.co.uk/scripts/latlong.html.

[2] Fire information for resource management system (firms). https://earthdata.nasa.gov/earth-observation-data/near-real-time/firms.

[3] F.A. Gers and J. Schmidhuber. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 3, pages 189–194 vol.3, July 2000.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[5] Michael I. Jordan. Chapter 25 - serial order: A parallel distributed processing approach. In John W. Donahoe and Vivian Packard Dorsel, editors, *Neural-Network Models of Cognition*, volume 121 of *Advances in Psychology*, pages 471–495. North-Holland, 1997.

[6] Benny Sofyan Samosir Nina Zakiah, Desniarti. Minimum spanning tree determination program using kruskal algorithm on visual basic 6.0. *International Journal of Science and Research*, pages 1817–1821, 2014.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Wikipedia contributors. Ball tree — Wikipedia, the free encyclopedia, 2022. [Online; accessed 11-February-2022].