Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		2412704

Build an Army of Drones to Fight Wildfires

Abstract

Global warming, El Niño...... With the emergence of various extreme climates,**Austral-ia's wildfires** occur more frequently. The greenhouse gases emitted after combustion have exacerbated global warming, which seems to have entered an endless loop. At the same time, hundreds of millions of lives have been killed in the fire, which makes us sad. In order to better control wildfires, we modeled the **distribution of drones** assisting in the observation to achieve the best balance between economy and efficiency.

Several models are established: Model I: Rasterized Multi-Objective Optimization Model; Model II: Model Verification Simulated by Poisson Process; Model III: Hovering Model Based on Tabu Search, etc.

For Model I: According to the **heat map** about distribution of fires in Victoria in recent years, we found that the main fire areas are the plains along the eastern coast. Inspired by weights and **Multi-Objective Optimization** algorithms, we built a brand-new model to find the best location for EOCs and draw up a suitable hover position and reconnaissance route for the drone. Based on the different positions of the fire site, calculate the maximum num-ber of two types of drones and their ratio. The results are shown in Figure 9.

For Model II: This model is actually a supplement to Model I. In the Model I, the fire only appeared in a small area, and there was a possibility of extreme fire events in the study area. Combining the data about nearly half a century and using the **Poisson Distribution** to obtain the probability and mathematical expectation, it can be concluded that 2.99431 ex-treme fire events may occur in the next ten years, which is approximately considered to be 3 times. After that, use the mobile EOC to deal with extreme fire events, and utilize the method of Model I to rebuild the drone network to find out what equipment costs need to be increased. Due to the diversity of the results, it will be shown in section 6.2.

For Model III: To tackle the problem of how to optimize the hover position of drones in different terrains, the **Tabu Search** algorithm (TS) is a good choice. Using the Tabu Search algorithm can make the hover position of the drones achieve the global optimal ef-fect under different terrain conditions. Since drone signals are severely interfered in urban areas, the reasonable distribution of EOCs enables it to quickly network to respond to sud-den urban fires. The obstruction of mountainous terrain restricts the drone's flying range. Therefore, dividing the area into blocks and managing them separately can effectively im-prove efficiency. $\frac{(x\cdot\gamma)\cdot\theta+8}{6\cdot\theta}$

Finally, sensitivity analysis of the mathematical expectation of extreme fire events ξ shows that our model is not sensitive to changes in ξ , that is, it can be applied to areas with different extreme fire invents. Meanwhile, robustness of our model has also been tested. While adding 5% random disturbance to d_{ki}^{α} and d_{ki}^{β} , the maximum time error is 3.2657%.

Keywords: Fighting Wildfires; Multi-Objective Optimization; Poisson Distribution; Tabu Search Algorithm; Sensitivity Analysis

Contents

1 Introduction

1.1 Background

If you are a fan in competitive sports games like tennis, you might have heard a lot of the word "Momentum". When a player has momentum, they seem to control the dominance of the match, wielding the match's rhythm and flow. Novak Djokovic, a record-breaking tennis player, described this in his saying: "... if you have the mental ability to stay strong, stay patient and confident and just have belief in the right moments, then you get a win, you know."

However, momentum is not stationary; you probably have witnessed momentum being lost or stolen from a player during a match, entirely changing the outcome. Momentum can switch to the other player within the blink of an eye and stun unsuspecting players and spectators. This is especially spectacular during the 2023 Wimbledon Gentlemen's final between Carlos Alcaraz and Novak Djokovic (Figure 1). In this match, Djokovic completely dominated the first set with a score of 6 – 1, displaying enormous momentum. However, this quickly shifted to Alcaraz in the third set, where he won 6 - 1, before Djokovic regained control in the fourth set with a 6 - 3 win. Finally, in the late stages of the fifth set, the momentum swung once again, and Alcaraz secured victory with a score of 6 - 4.

But where **exactly** did that momentum come from? Despite its recognized impact, momentum has remained largely a qualitative factor and rooted in the subjective experience of spectators and self-perception of athletes. The question, then, is whether this intangible force can be analyzed and systematically understood through mathematical methods. If this proves feasible, it has the potential to greatly enhance the strategies used by competitive tennis players and coaches and, by extension, athletes and coaches in various sports.

Figure 1: Carlos Alcaraz (Left) and Novak Djokovic (Right) at Wimbledon [1]

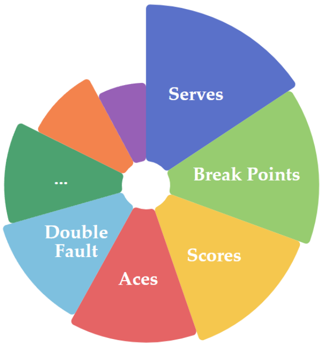


Figure 2: Data Aspects Provided by *Wimbledon_featured_matches.csv*

1.2 Problem Analysis

Momentum is a *phenomenon*, which means it is an outcome of a wide range of factors, including psychological states, like confidence and mental toughness, to external conditions like the current state of the match. To analyze and build a robust model that measures the momentum of players in competitive tennis, it is crucial to cover these factors extensively. Specifically, our model should be capable of completing the following tasks:

(1) Evaluate the quatitive degree how one player is performing better than the other player at any given point in the match, and illustrate the changes of such degree visually.

(2) Based on extensive factors, including the flow of match, the model should quantify the concept of "momentum" in tennis matches, and identify potential indicators that could predict shifts in momentum. To address the coach's skeptism, the model needs to be compared to a null model that assumes outcomes are random (e.g., based on point-by-point win probabilities without momentum).

(3) Finally, the developed "momentum" model should be tested on other matches from the different matches, including women's matches, to exhibit its predictive accuracy and generalizability, and identify any limitations or additional factors that may need to be incorporated.

1.2.1 Capturing the Match's Flow

To compare the performance ratings between the two players, we can use the data provided in *Wimbledon_featured_matches.csv* that describe the explicit match information, such as whether a player was serving or at a break point, etc. (shown in Figure 2); but we also need to consider other implicit information that has significant impact on the state of the match, for example, whether a player has a high scoring streak. Other unimportant information, potentially the speed of the serve, needs to be neglected to ensure model's simplicity and robustness. This inherently requires **feature engineering**: to extract and construct feature variables that capture both explicit and implicit aspects that control the flow of the match.

Since our model aims to dynamically assess the match's state as it evolves, it requires a granular focus on the ebb and flow of the game, taking into account both minor and major developments at any given moment. For example, the player may have an advantage in games won, but when the other player just scored a break point, that player could be performing much better at that specific time. Therefore, to scope the data to recently played points, the **sliding window** technique can be employed; to handle older scores and progressively update the model as the match goes on, introduction of a **forgetting factor** would be applicable, as illustrated in Figure 3. The remaining work would be finding an appropriate model to map these parameters to the performance or the win rate of a player.

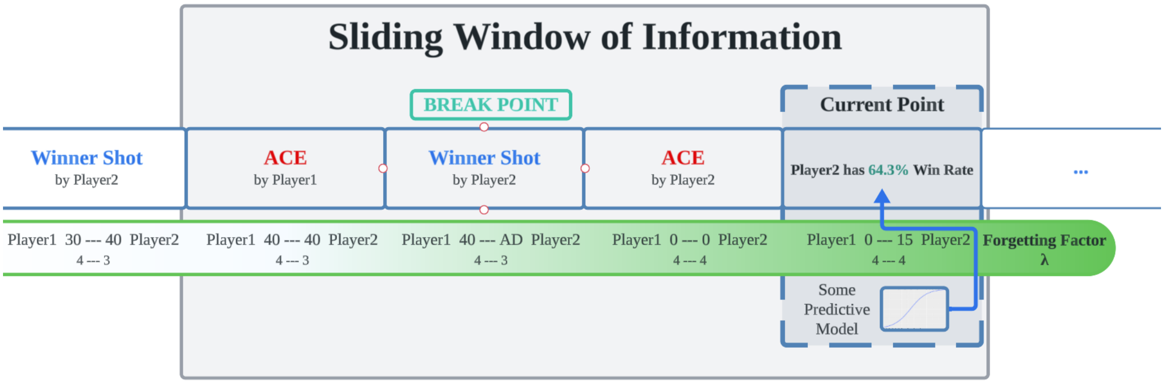

Sliding Window of Information
BREAK POINT
Current Point
Winner Shot by Player2
ACE by Player1
Winner Shot by Player2
ACE by Player2
Player2 has 64.3% Win Rate
...
Player1 30 --- 40 Player2 4 --- 3
Player1 40 --- 40 Player2 4 --- 3
Player1 40 --- AD Player2 4 --- 3
Player1 0 --- 0 Player2 4 --- 4
Player1 0 --- 15 Player2 4 --- 4
Forgetting Factor λ
Some Predictive Model

Figure 3: Information Is Constrained by Sliding Window and Forgetting Factor. (Note that the data in this graph is for display purposes only.)

1.3 Literature Review

This question is mainly about mobilizing cluster drones to extinguish wildfires. In re-cent years, research on optimization algorithms for UAV(drone) cluster path planning is very hot, Generally, it can be divided into two parts, the **Swarms of UAVs' Path Planning Model** and the **Swarms of UAVs' Path Planning Optimization Algorithm**, this section mainly discusses the models that have been proposed.

- First of all, in terms of the dimensionality of the space: In[2] HU et al. sets the plan-ning space to three dimensions. However, in order to simplify the model, more authors tend to consider the space as two dimensions[3].
- Secondly, in terms of the method of planning space: the commonly used methods in-clude Grid Method[4], Road Sign Method, and Artificial Potential Fields Method.
- Finally, the objective function of the UAV Cluster Path Planning Model generally uses flight distance, threat cost, etc. For example, Xu et al.[5] takes the weighted sum of threat cost and time cost as the optimization target. What's more, Constraints often in-clude self-constraints and environmental constraints, such as flight speed and geo-graphic altitude.
- The strengths and weaknesses of the planning space can be visually presented and is shown below:

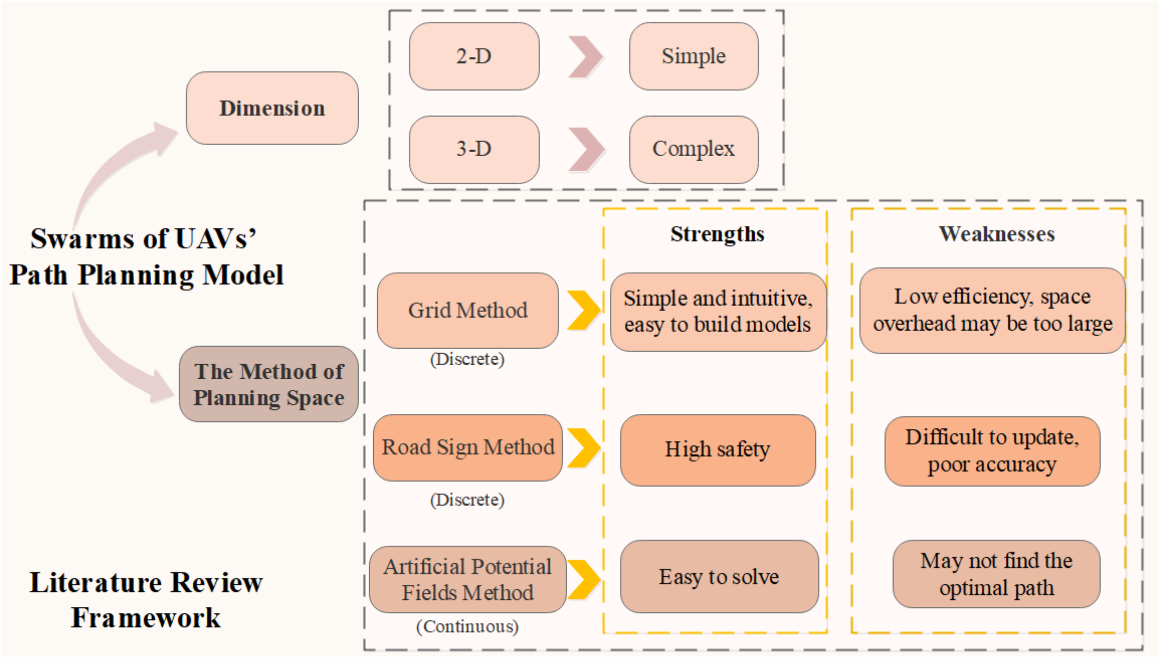

Dimension
2-D
Simple
3-D
Complex
Swarms of UAVs' Path Planning Model
The Method of Planning Space
Strengths
Weaknesses
Grid Method
(Discrete)
Simple and intuitive, easy to build models
Low efficiency, space overhead may be too large
Road Sign Method
(Discrete)
High safety
Difficult to update, poor accuracy
Artificial Potential Fields Method
(Continuous)
Easy to solve
May not find the optimal path
Literature Review Framework

Figure 4: Literature Review Framework

1.4 Our Work

In this paper, we seek to mathematically model the flow of momentum across a tennis match, and use it to provide insights to benefit tennis coaches and atheletes across the world. It is our belief that a practical and inclusive mathematical framework would greatly help the tennis community in understanding the concept of momentum to improve performance, strategy, and training methodologies. Through extensive investigation and research in the field of tennis, our main model, the **Dynamic Tennis Momentum Analyzer (DTMA)**, integrates a comprehensive set of features and use a variety of statistical methods to achieve this goal. The building of this model can be overall described as follows:

(1) To capture the flow of match in real-time, we constructed the CourtSense model as a submodel of DTMA. The model starts by applying a Sliding Window machanism to a set of extracted features; these features are passed into the core Progressive Logisic Model that use the set of principal factors to predict the winning chance of the point; due to the real-time progression of the competition, a Forgetting Factor is added to dynamically update this model to reflect the most recent match status.

(2) The mixture of the two drones is given and the extreme fire events is considered;

(3) Based on the verification model simulated by Poisson process and the hovering model based on Tabu Search, this article effectively demonstrates the validity and applicability.

In order to avoid complicated description, intuitively reflect our work process, the flow chart is shown in Figure 3:

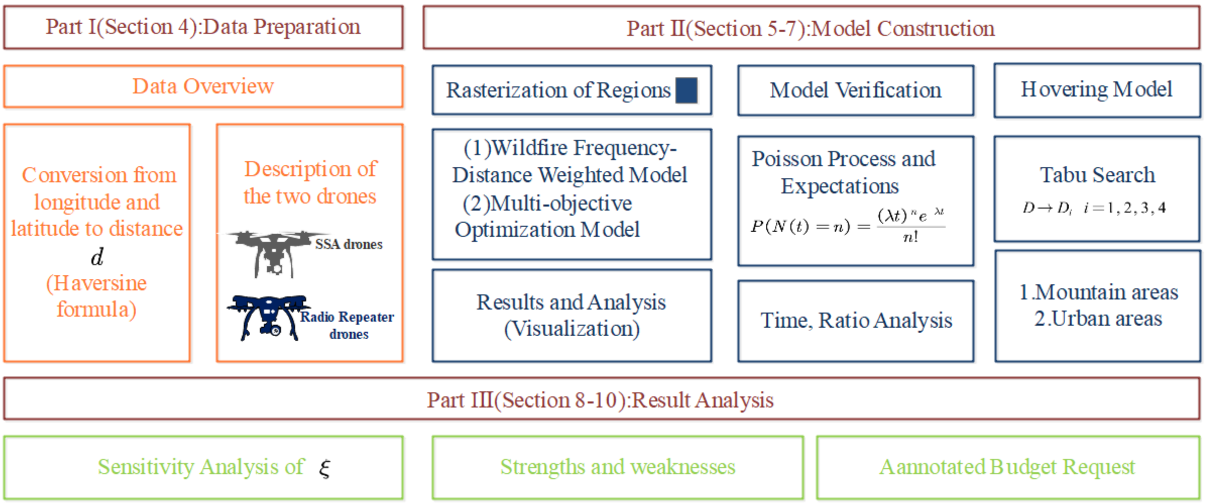

Part I(Section 4):Data Preparation
Part II(Section 5-7):Model Construction
Data Overview
Rasterization of Regions
Model Verification
Hovering Model
Conversion from longitude and latitude to distance
d
(Haversine formula)
Description of the two drones
SSA drones
Radio Repeater drones
(1)Wildfire Frequency-Distance Weighted Model
(2)Multi-objective Optimization Model
Poisson Process and Expectations
$P(N(t)=n)=\frac{(\lambda t)^n e^{\lambda t}}{n!}$
Tabu Search
$D\rightarrow D_i \quad i=1,2,3,4$
Results and Analysis (Visualization)
Time, Ratio Analysis
1.Mountain areas
2.Urban areas
Part III(Section 8-10):Result Analysis
Sensitivity Analysis of ξ
Strengths and weaknesses
Aannotated Budget Request

Figure 5: Flow Chart of Our Work

2 Assumptions and Explanations

Considering that practical problems always contain many complex factors, first of all, we need to make reasonable assumptions to simplify the model, and each hypothesis is closely followed by its corresponding explanation:

Assumption 1: **Only the influence of terrain on the drone is considered, and other factors such as temperature, humidity, and atmosphere are ignored.**
Explanation: The UAV's radiation communication range is only affected by terrain factors, and other factors are minimal. In fact, these factors affect each other, but in order to simplify the model ,we ignore the interactions between these factors.

Assumption 2: **The location of the EOC can be deployed around the fire due to an emergency.**
Explanation: The position of the EOC is not clearly given in the question. So, we assume that EOC can be set in a fire-free area around the wildfire site, since the glossary of the problem said that mobile EOC can be deployed near the site of an emergency.

Assumption 3: **The "boots-on-the-ground" forward teams can be approximated as near the fire site.**
Explanation: The actual movement of the team is very complicated, and it is difficult to accurately calculate its position. Therefore, it is assumed that the squad is near the fire field and the drone arrives at the fire field to establish a connection with the squad.

Assumption 4: **The collected data can be considered reliable and can reflect the changing laws of the Victorian wildfire.**
Explanation: The historical Victorian wildfire data, latitude, longitude and other data come from authoritative websites, such as the official website of FEC in Australia and NASA, with high accuracy.

Additional assumptions are made to simplify analysis for individual sections. These assumptions will be discussed at the appropriate locations.

3 Notations

Some important mathematical notations used in this paper are listed in Table 1.

Table 1: Notations used in this paper

Symbol	Description
x_i	Longitude within the i-th Wildfire Grid
y_i	Latitude within the i-th Wildfire Grid
$\mathit{\Omega}_i$	The area of the i-th grid
d_{ki}	the distance d_{ki} between the k-th roaming grid and the i-th grid
SC_k	Score for evaluating the k-th wildfire grid
$x_{ki}^{(\alpha)}$	the SSA_α drone sent by the k-th EOC to the i-th wild-fire grid
$x_{ki}^{(\beta)}$	the RR_β drone sent by the k-th EOC to the i-th wildfire grid
t_{fly}^{δ}	The flight time of drones

*There are some variables that are not listed here and will be discussed in detail in each section.

4 CourtSense Model: Judging the State of the Match

The CourtSense model aims to capture the current state of the match by predicting the outcome of the next unplayed point, using available match information. First, we need to apply feature engineering to the given data to extract useful features.

4.1 Feature Engineering

4.1.1 Data Extraction

The data provided by the problem (*Wimbledon_featured_matches.csv*) is a comprehensive dataset that contains the complete match information, including scores, serve details, point details which are all applicable to analyze the state of the match. After a thorough inspection of the data, we find that the data is clean and usable. However, these are all explicit match information and is limited to a specific point, for example, whether this player is serving, what is the ratio of scores, etc. We think that more temporal features like whether this player is leading in score are more direct in describing the match's status; hence we added them to the feature set. These additional features are included in Table 2.

These features, representing nuanced indicators of the current state of the match, are better to feed into the model than raw scores for simplicity and accuracy. For example, the winning streak is a direct reflection of the player's current performance and momentum.

Table 2: Added Match Information

Column Name	Description	Example Value
p1_streak	The winning / losing streak of player 1 up until that point	5, 0, -3
p2_streak	The winning / losing streak of player 2 up until that point	3, 0, -5
p1_leading	Whether p1 is leading in score	1 (leading), 0, -1
p2_leading	Whether p1 is leading in score	-1 (losing), 0, 1
p1_leading_game	Whether p1 is leading in game	1, 0, -1
p2_leading_game	Whether p1 is leading in game	-1, 0, 1
p1_leading_set	Whether p1 is leading in set	1, 0, -1
p2_leading_set	Whether p1 is leading in set	-1, 0, 1

Moreover, for the data columns like '*game_victor*' that represent the winner of the game (or set), we split them into 2 columns to be player-specific. For each point i in the match, we transform, for example, the *game_victor* column into two player-specific columns, $V_{p1}^{(i)}$ and $V_{p2}^{(i)}$, defined as:

$$V_{p1}^{(i)} = \begin{cases} 1 & \text{if } \textit{game\_victor} = 1 \text{ at point } i \\ -1 & \text{if } \textit{game\_victor} = 2 \text{ at point } i \\ 0 & \text{if if } \textit{game\_victor} = 0 \text{ at point } i \end{cases}$$

$$V_{p2}^{(i)} = \begin{cases} 1 & \text{if } \textit{game\_victor} = 2 \text{ at point } i \\ -1 & \text{if } \textit{game\_victor} = 1 \text{ at point } i \\ 0 & \text{if if } \textit{game\_victor} = 0 \text{ at point } i \end{cases}$$

this procedure ensures these data can be modeled as their respective performance.

Finally, we discarded other rather unimportant or redundant information for the match, such as the raw score information, the speed of the serve (which cannot be used for points already in play when predicting point outcomes), unforced errors (not an indicator of player's performance), and information about net points. These choices are made after conducting a thorough research on related Sports Analytics in the field of tennis [2][3][4]. The remaining features alongside the constructed features can be used as raw input data for our model.

4.1.2 Sliding Window Mechanism

We recognize that the current state of the match is heavily influenced by the temporal features of, not only for the current point, but also the last few points. For example, if the player ran a long distance in the past few point rounds, they will carry more amount of fatigue, influencing the player's performance of the current point. Or if the player has just won a break point, they will gain much more confidence in their coming plays. Therefore, by employing a sliding window mechanism, the model can utilize the temporal features on the most recent and relevant data points, ensuring that the analysis remains sensitive to the current state of play and the immediate history before it. This is described in the formula below:

Let F_t and F_{nt} represent the sets of temporal and non-temporal features, respectively. For a given point in time t, the combined feature set $F^{(t)}$ is given by:

$$F^{(t)} = S_t^{(t-1)} \cup F_{nt}^{(t)}$$

where

$$S_t^{(t-1)} = \sum_{i=t-3}^{t-1} F_t^{(i)}$$

represents the sum of temporal features of last 3 points played, and F_{nt} represents the non-temporal features at the current point t. So far, we have acquired a granular and exhaustive features that completely describe the match's status at a specific point and is ready to be fed into our Progressive Logistic Model - the core of the CourtSense model.

4.2 Progressive Logistic Model

To effectively capture the dynamic flow of a tennis match, we introduce the Progressive Logistic Model. This model is designed to predict the outcome probability of the current point, providing a real-time assessment of the match's status and the relative performances of the players. The Progressive Logistic Model utilizes a multivariate logistic regression framework, due to its suitability for binary outcome prediction (e.g., point won or lost) and its ability to output probabilities based on multiple features; furthermore, combined with the Forgetting Factor, the model is capable of progressively updating the model according to the latest match information as the match progresses. The general process of this model is depicted in Figure 3.

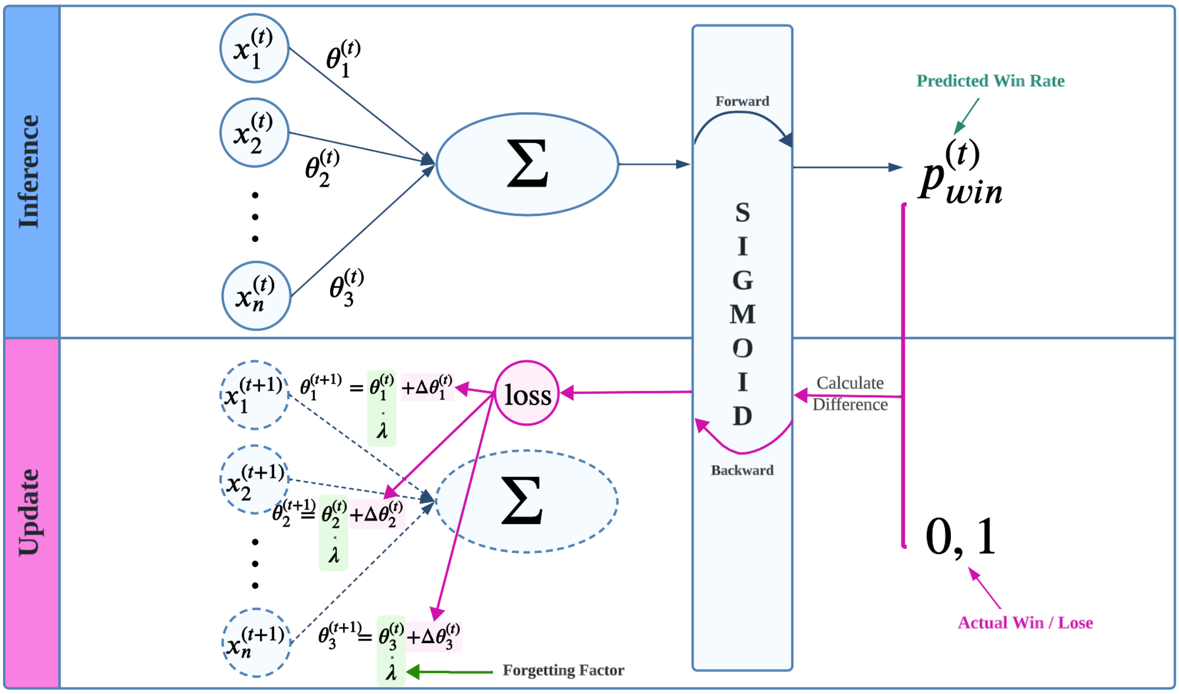


Figure 6: Process of the Progressive Logistic Regression Model

4.2.1 Model Establishment

Logistic regression is a statistical model used for binary classification problems, developed by David Cox in 1958. It is a type of generalized linear model (GLM) that uses a logistic function to model a binary dependent variable. The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = 0 or 1) and a set of independent variables (predictors) [5]. While the assumption of variable independence is ideal, logistic regression can still perform effectively in real-world scenarios where complete independence is rare, such as our feature set that describe match's information.

To feed the aggregated features into a logistic regression model, each feature set is represented as a vector $x^{(t)}$ at timestep t (which means the t-th point played in this match), and these vectors are used to calculate the probability of winning a point. $x^{(t)}$ is directly derived from the feature set $F^{(t)}$, which contains both temporal features and non-temporal features.

The logistic regression model predicts the probability $P(y^{(t)} = 1|\mathbf{x}^{(t)}; \theta)$ where $y^{(t)} = 1$ indicates the event of interest (i.e., winning the point), $x^{(t)}$ is the feature vector at timestep t, and θ represents the model parameter (weights). The model calculates the probability using the formula below:

$$P(y^{(t)} = 1|\mathbf{x}^{(t)}; \theta) = \sigma(\theta^\top \mathbf{x}^{(t)})$$

where $\sigma(z)$ is the sigmoid function defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

and $z = \theta^\top \mathbf{x}^{(t)}$ represents the linear combination of the features $\mathbf{x}^{(t)}$ weighted by the model parameters θ, expressed as:

$$z = \theta_0 + \theta_1 x_1^{(t)} + \theta_2 x_2^{(t)} + \cdots + \theta_n x_n^{(t)}$$

where θ_0 is the intercept term, $\theta_1, \theta_2, \ldots, \theta_n$ are the weights associated with each feature $x_1^{(t)}, x_2^{(t)}, \ldots, x_n^{(t)}$ in the feature vector $\mathbf{x}^{(t)}$.

Once we have the correct set of model parameters, we can determine the winning probability of a player based on the feature vector of the match for that player.

4.2.2 Parameters Training & Updating

One of our Progressive Logistic Model's highlights is its ability to progressively adjust itself during the play of the match. But this does not mean it relies only on the real-time data; instead, we first train the model using the data given by the problem (pre-training phase), and when applied to a specific match, the progressive real-time data updates the parameters as the match unfolds (updating phase).

The training of the logistic regression model's parameters θ is achieved by minimizing the logistic loss function, which quantifies the difference between the predicted outcomes and the actual results of matches:

$$\mathcal{L}(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(\sigma(\theta^\top \mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - \sigma(\theta^\top \mathbf{x}^{(i)})) \right]$$

where m represents the number of observations in the training dataset, $y^{(i)}$ denotes the outcome of the i-th point (win or loss), and $\mathbf{x}^{(i)}$ is the feature vector associated with the i-th point in a match. Then, the parameter vector θ is optimized performed using Gradient Descent, described as:

$$\theta := \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$$

where α is the learning rate that regulates the step size during each iteration, and $\nabla_\theta \mathcal{L}(\theta)$ represents the gradient of the loss function with respect to the parameter vector θ. Considering the constraints of the size of the dataset and the model's ability to update itself during a match, we meticulously choose different values for pre-training our model and updating our model during the match:

$$\alpha = \begin{cases} 0.01 & \text{at pre-training phase} \\ 0.05 & \text{at updating phase} \end{cases}$$

in this way, for each data point, both when pre-training and when updating, the parameter vector is iteratively updated in the direction that reduces the loss function.

At updating phase, the logistic regression model's parameters (θ) need to reflect the latest match data. This requires us to place more weight to the latest information of the points. Therefore, aside from using a higher learning rate value, a forgetting factor λ is incorporated into the parameter update rule:

$$\theta^{(t)} = \lambda \cdot \theta^{(t-1)} + \Delta\theta^{(t-1)}$$
$$\Delta\theta^{(t-1)} = -(1-\lambda)\alpha\nabla_{\theta^{(t-1)}}\mathcal{L}(\theta^{(t-1)})$$

this rule adjusts $\theta^{(t)}$ by blending the previous parameters $\theta^{(t-1)}$, scaled by λ, with the changes updated by new data ($-\alpha\nabla_{\theta^{(t-1)}}\mathcal{L}(\theta^{(t-1)})$), scaled by $(1-\lambda)$. In such a way, the forgetting factor allows the model to prioritize recent information and adapt to changes in gameplay strategies or player performance over time.

To this point, we have thoroughly established our CourtSense model, which first uses feature selection combined with a sliding window mechanism, and leverages a Progressive Logistic Model to follow the flow of the match and predict the result of a specific point.

The original dataset of *Wimbledon_featured_matches.csv* included the matches played at Wimbledon 2023 from the 3rd round to the final round (7th round). To separate the training and testing sets, we use the data of the 30 matches from the 3rd round to the semi-final round, and use the epic match in the final round between Carlos Alcaraz and Novak Djokovic to validate our model's accuracy.

We train our CourtSense model using Scikit-learn, a machine learning framework for Python. After pre-training, the CourtSense Model is then tested on the testing set, where the model starts from predicting the winning probabilities of both players at the first point, and updates itself based the actual result of the first point, and then moves on to predict the second point, etc. The aggregated results of all point predictions and their actual outcomes are used for validating our model's performance.

4.3 Model Validation

In validating the performance of our CourtSense model, we focused on a suite of standard evaluation metrics, including Mean Squared Error (MSE), accuracy, confusion matrix, and AUC-ROC. MSE represents how far the model's predictions are from the actual values; accuracy measures the proportion of total predictions our model gets right; the confusion matrix shows exactly how many values were predicted correctly or falsely; AUC provides an insight into the model's ability to distinguish between the two possible outcomes (winning or losing a point), with higher values close to 1 indicating better performance. The detailed explanation can be found in [6]. The result of these metrics are shown in Table 3 and Figure 7.

Table 3: Evaluation Metrics Result for CourtSense Model

Metric	Value for Player 1	Value for Player 2
MSE	19.57%	19.02%
Accuracy	92.52%	93.11%
AUC-ROC	93.886%	93.445%

To our surprise, the CourtSense model performs exceedingly well at predicting each point's outcome during the match. It is almost too good: the model achieved an average accuracy of 92.82%, while the AUC stood at an average of 93.67%, suggesting that the model can reliably

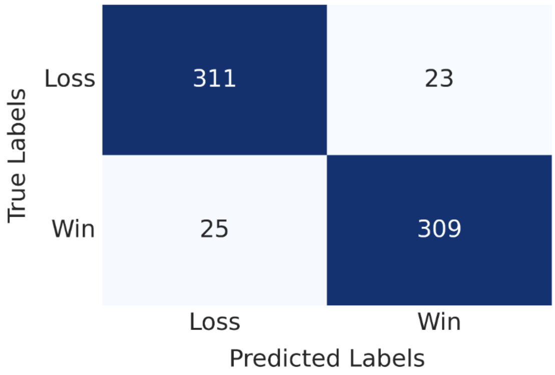


Figure 7: Confusion Matrix for CourtSense Model

forecast match dynamics, providing valuable insights into players' performance. This could be attributed to our meticulous selection of features and the various mechanisms our model used.

In addition to evaluating the overall performance of our CourtSense model, we also analyzed on its adaptability throughout the progression of a match. We analyzed the model's accuracy in predicting outcomes by computing a moving average across 20-point windows. This approach allowed us to observe how the model's predictive accuracy fluctuated over the course of a match. The resulting data is illustrated in Figure 8.

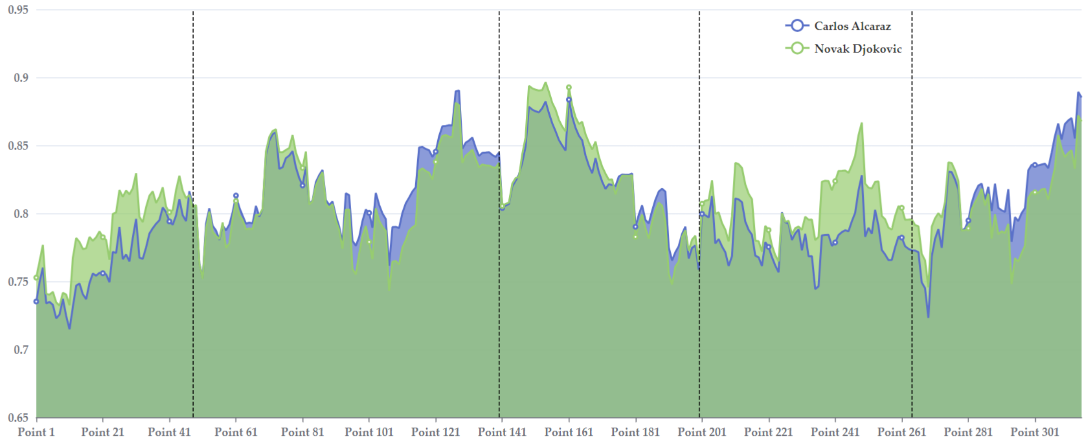


Figure 8: Average Predictive Power of CourtSense as Match Unfolds

We can see the model starts at a relatively low accuracy, and moves up as the match unfolds. Generally, for each set of the game (as separated by the vertical lines in the graph), the model is able to progressively adapt its parameters to better fit the match's status, although there are some fluctuations, probably signaling a randomness for the outcome of the points.

4.4 Match Flow Visualization

To visualize the flow of the match, we select the 2nd set of the final match between Carlos Alcaraz and Novak Djokovic, which was an intense play with a tie-breaker of 7-6. Figure 9 presents a graphical depiction of the prediction difference between the two players, which indicates their performance differences, over the course of this set. The vertical axis is the difference in the predicted probability of winning each point; while the horizontal axis represents the sequence of points played.

We can see the fluctuations in the graph that trace the ebb and flow of the match. When the curve rises above the zero line, it suggests that Player 1 is more likely to win the point; when it dips below, Player 2 is favored. The dotted line at 0.08 reflects the average prediction difference across all points, which is a little above 0 (happens to reflect the tie-breaker win of Alcaraz).

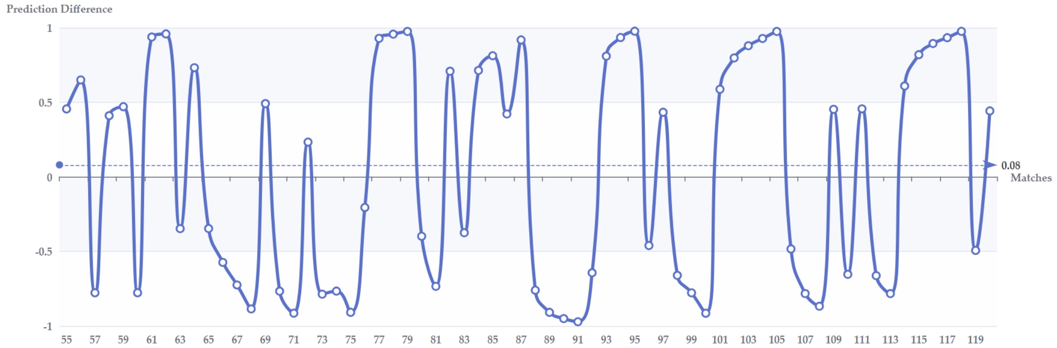
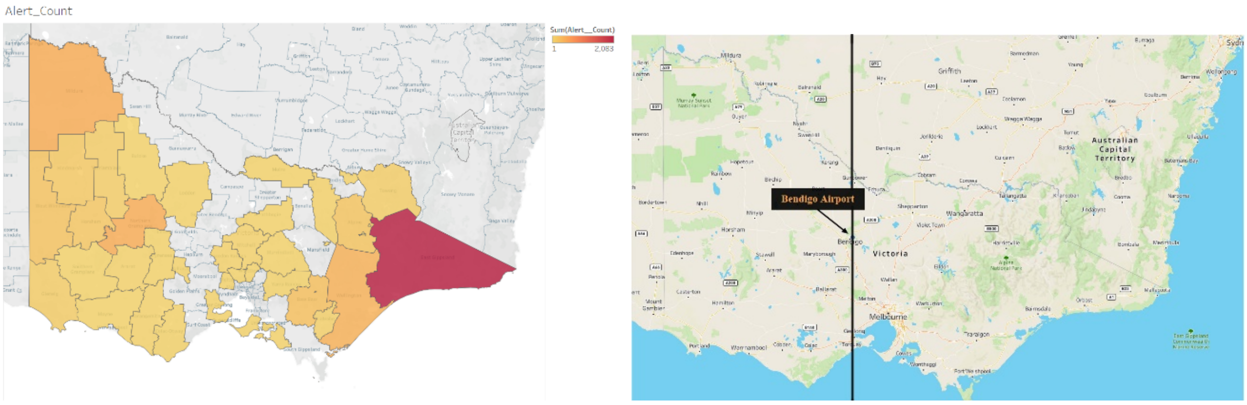


Figure 9: Match's Flow during the 2nd Set of the Final Match, Indicated by the Winning Chance Difference

4.4.1 Data Screening

Judging from the map of Victoria in Figure 4 (right), the eastern region is mainly forest, while the western region is almost no forest. Furthermore, to demonstrate better the situa-tion of wildfires, we plot over a heat map in Figure 4 (left). Considering the heat map we made, it shows the number of wildfires in various states of Victoria from 2012 to 2021, the darker the color, the greater number of fires. Although fires have also occurred in the western region, the number of eastern regions is much higher than that in the western region.



(a) Data Screening(left)

(b) Data Screening(right)

Figure 10: Data Screening

1. The analysis of locations of hovering VHF/UHF radio-repeater drones for fires can be more accurate if we have more complete data;
2. The assumption that the "boots-on-the-ground" forward teams can be approximated as near the fire site is a bit idealized. If the trajectory of the team is taken into account, a more practical model and results can be obtained.
3. Some approximate analysis methods are applied to model other places, which may lead to the situation that not to be the most optimal.

$$\frac{(x \cdot \gamma) \cdot \theta + 8}{6 \cdot \theta} \tag{1}$$

Algorithm 1 BLAH BLAH BLAH

```
Require: wordlist, gameround, v, r, f
Ensure: bestguess
$pguesses \leftarrow []$
$freq \leftarrow modfreq(wordlist)$
for $i \leftarrow 1$ to $length(wordlist)$ do
    $w \leftarrow weightings(wordlist[i], freq, v, r, f)$
    $pguesses.append([w, wordlist[i]])$
end for
$pguesses.sort(reverse = True)$
return $pguesses[0][1]$
```

Table 4: Data and Database Websites

Database Names	Database Websites
Fire Alerts	https://www.globalforestwatch.org/map/
Altitude	https://search.earthdata.nasa.gov/search
Latitude and Longitude	https://www.kaggle.com/carlosparadis/
Google Scholar	https://scholar.google.com/
Maps	© 2021 Mapbox © OpenStreetMap

References

[1] Skysports | Wimbledon: Carlos Alcaraz dismantles Daniil Medvedev to set up final against Novak Djokovic.

[2] Medium | Predicting ATP Tennis Match Outcomes Using Serving Statistics.

[3] Avinash Dixit, Susan Skeath. The Most Important Situations in Tennis – and in R&D Competition.

[4] Pual K. Newton, Joseph B. Keller. Probability of Winning at Tennis I. Theory and Data.

[5] JS Cramer, University of Amsterdam, 2022. The Origins of Logistic Regression.

[6] Analytics Vidhya | Metrics to Evaluate your Classification Model.

A Appendix I: The Genetic Algorithm

blah,blah,blah

B Appendix II: Mail to the S.H.I.E.L.D

blah,blah,blah

Letter to Central Intelligence Agency

To: Recipient Name
From: Your Name
Subject: The Subject of Your Letter
Date: February 4, 2024

We want to develop and summarize a model to classify solution words by difficulty.We constructed an AI simulating human thinking to solve problems based on iterative optimization parameter finding strategy. It could solve wordle about 3 times on average. We trained 450 such AI and took the average number of solving problems as the evaluation standard of difficulty. After that, we used k-means clustering algorithm to successfully classify normal problems and difficult problems, and found that normal problems generally have the characteristics of high word frequency, small number of non-repeating letters and so on.We also used correlation analysis to assess the relationship between the ai difficulty prediction and the average number of problems solved by humans, which showed a significant correlation. This verifies the accuracy of our model. Finally, we visualized the data and found an interesting conclusion: although daily reported results decreased, the proportion of difficult patterns increased. We attribute this to the presence of strong old players and a reluctance to post simple reports.We also found that the number of non-repeating letters also affected the percentage of hard mode. That's all from our data analysis. I hope it's helpful to your game. Thanks for watching.

Report On the Use of AI

1. OpenAI *ChatGPT* (Jan 10, 2024 version, ChatGPT-4)

Query1:

这个为什么没有 Appendices 标识

Output:

在 LaTeX 中，默认情况下，`\appendix` 命令并不会自动添加一个 "Appendices" 的标题或者标识。`\appendix` 命令主要是用来改变章节编号的格式，让其从接下来的部分开始用字母代替数字进行编号（比如 "附录 A", "附录 B" 等）。如果你希望在附录部分之前有一个明确的 "Appendices" 标题或者标识，你需要手动添加它。

这里有一个简单的方法来添加 "Appendices" 标题：

```latex
\documentclass{article}

\begin{document}

\section{主要部分}
这里是你文档的主要部分。

% 添加一个不编号的章节作为附录的标题

\section*{Appendices}

% 以下命令确保附录部分的章节编号以字母形式

\appendix
```

```
\section{附录A}

这里是附录A的内容。

\section{附录B}

这里是附录B的内容。

\end{document}
```

在这个例子中，`\section*{Appendices}`创建了一个不编号的章节标题"Appendices"。接下来的`\appendix`命令确保了之后的章节将以字母编号（即作为附录部分）。

请注意，这个例子适用于`article`文档类。如果你使用的是`report`或`book`文档类，你可能需要使用`\chapter*{Appendices}`代替`\section*{Appendices}`，因为这些文档类支持`\chapter`命令。

这种方法提供了一种简单的方式来明确标示文档的附录部分，同时保持LaTeX文档的结构和格式的一致性。

2. OpenAI *ChatGPT* (Jan 10, 2024 version, ChatGPT-4)

Query2:

1234

813232

123r2r43

Output:

It seems like you've entered a series of numbers and a combination of numbers and letters. If these are meant to be codes, passwords, or specific identifiers, I'll need more context to understand their purpose or how to assist you with them. If this is related to a programming, encryption, or a similar technical challenge, please provide more details so I can offer a comprehensive response. If it's something else, like a puzzle or a specific reference to a topic, let me know so I can address it appropriately.