CHAPTER 2

LITERATURE REVIEW

2.1 Research Background of Wildfire Prediction and Evolution of Data Model Methods

In many wildfire-related studies, researchers have tried to figure out what kinds of conditions might lead to the start or spread of a fire. These conditions can come from different areas—some related to weather, some tied to the physical landscape, and others that seem to be more connected to human activity. Although these factors are often studied separately, in reality, they usually work together in ways that are hard to fully separate.

For example, temperature, rainfall, humidity, and wind speed are all commonly used variables in fire prediction. These are often referred to as meteorological factors. When it gets too hot and dry, the chances of vegetation catching fire seem to increase. Wind, on the other hand, may help spread a fire faster once it starts. But it's not always that simple—sometimes, even with dry conditions, fires don't break out unless other triggers are present.

Then there's terrain. Features like elevation, slope, and aspect (the direction a slope faces) can influence how easily a fire moves through a landscape. Steeper slopes might speed up fire spread, while valleys or flat areas may slow it down. Some studies also mention that areas facing south or west tend to be drier, which might raise fire risk—but this can depend a lot on the region.

Vegetation conditions are another piece of the puzzle. Variables such as NDVI (Normalized Difference Vegetation Index) or leaf moisture content are often used to estimate how dry or flammable the plants in an area might be. When vegetation is dry or dead, the risk of ignition and spread seems to go up. Still, how exactly vegetation interacts with other factors isn't always clear—it probably changes by season and location.

Lastly, human activity is also considered important. Things like population density, distance to roads, or land use type can all play a role. Fires are sometimes caused by people—either accidentally or on purpose—and the way land is developed may increase how likely fires are to start or spread. For instance, building communities near forests or in wildfire-prone zones may make those areas more vulnerable.

So overall, wildfire risk is shaped by a mix of different factors, and it's hard to point to just one cause. Most studies suggest that these variables work together in complicated ways, and that fire prediction usually works best when models are built using several types of features rather than relying on a single one.

2.2 Data Processing and Feature Construction

Before training a prediction model, it's usually necessary to do quite a bit of data preparation. That's because wildfire-related data often comes from many different places, and the formats, time intervals, or coordinate systems don't always match. So, to make everything usable, researchers often go through several steps to clean and process the data. These steps might not be exactly the same in every study, but there are some common patterns that show up pretty often.

The first task is usually handling missing values. This happens when, for example, part of a temperature record is missing for a certain day, or a location doesn't have wind data. Depending on how much data is missing, researchers may fill the gaps using the average, use values from nearby time periods, or sometimes just remove the sample altogether if the missing portion is small. The main goal here is to make sure each data point used for training has all the needed variables, so the model isn't affected too much by incomplete records.

After that, it's important to line up the data in time and space. Since some variables—like weather data—are recorded hourly, and others—like fire occurrence—are recorded daily or weekly, researchers often resample or average the values to match the same time scale. Spatial alignment is also needed because data from different sources may use different resolutions or map projections. To fix this, methods like resampling or coordinate transformation are usually applied, which helps ensure that all features refer to the same area at the same time.

Once everything is cleaned and matched, the next step is deciding which variables to include. This part is often called feature selection. In many studies, people choose variables based on domain knowledge, past research, or even trial-and-error. Things like high temperature, low humidity, and strong wind are often seen as signs of higher fire risk. Variables like NDVI (which gives a rough idea of how green or dry the plants are) are also popular, since vegetation health seems to affect fire behavior. In some cases, researchers also consider human-related factors, such as population density or land use, especially if they're studying regions where human activity plays a big role.

Apart from selecting features, some studies also try to create new variables from the existing ones. This is called feature construction or feature engineering. For example, some researchers calculate a dryness index by combining temperature and

humidity. Others might create an interaction variable between slope and wind speed to represent how fires could spread uphill. These new features try to reflect how different conditions interact in real life, and sometimes they help the model find patterns more easily.

One final step that's often used before modeling is scaling the features. That's because different variables can have very different ranges—like temperature might be between 0 and 40°C, while population density could be in the hundreds or even more. If these values are used directly, variables with larger ranges might dominate the model's learning process. To avoid that, methods like normalization (min-max scaling) or standardization (Z-score) are commonly applied to put everything on a similar scale.

Since this is a supervised learning task, labels also need to be created. In wildfire prediction, labels usually indicate whether a fire occurred in a given area during a certain time. It's usually a binary value—1 for fire, 0 for no fire—based on historical fire records. Once the features and labels are ready, researchers typically split the dataset into a training set and a test set. The training set helps the model learn, and the test set checks how well it performs on new data.

To sum up, the process of cleaning, organizing, selecting, and constructing features is an essential part of building a good wildfire prediction model. If the data going in is messy, the model probably won't work well—no matter how advanced the algorithm is. But if the data is prepared thoughtfully, the model may be more likely to find useful patterns and make better predictions.

2.3 Modeling Methods for Wildfire Prediction

2.3.1 The application of traditional supervised learning models in wildfire prediction

In wildfire prediction research, one common approach is to set up the task as a supervised learning problem. This usually means using past records that contain environmental data ─ like temperature, rainfall, wind speed ─ and labels showing whether a fire happened in that place and time. The goal is to help the model figure out what kinds of patterns might be linked to fire occurrence, so that it can later make predictions in similar situations.

At first, researchers mostly worked with models that had relatively simple structures. These traditional machine learning methods were popular partly because they were easier to understand and didn't need much computing power. For example, logistic regression has been widely used to estimate the probability of a fire based on different input variables. It's straightforward to apply, and its results can usually be interpreted without too much difficulty. But one common issue is that it assumes the relationship between features and the outcome is linear, which might not fully reflect how fires actually behave in nature. In many real-world cases, the interactions between variables are probably more complex than what a basic regression model can capture.

Because of this, many researchers began trying models that could better handle non-linear relationships. Decision trees are one of the options that have been used. They work by splitting data into branches based on conditions—like "Is the temperature higher than 30°C?"—and these branches lead to predictions at the end points. Decision trees work by splitting the input data into different branches based on

conditions—like temperature, wind speed, or other variables—and eventually making a prediction at the end of each path. They're often used because they can pick up on more complex relationships between variables, and the way they operate is still relatively easy to follow. That said, they can be pretty sensitive to noise in the data. If the model ends up focusing too much on patterns in the training data, it might not do as well when it's tested on new information—a situation that's usually referred to as overfitting.

To avoid this problem, researchers sometimes use ensemble methods like random forests. Rather than building a single decision tree, a random forest builds many of them, each trained on slightly different samples. The idea is that by letting all the trees "vote" on the result, the model becomes more stable and less likely to overfit. This approach has been applied in a number of wildfire prediction studies, and it seems to work especially well when the dataset includes lots of variables and enough samples to train on.

Another method that shows up quite a bit is the support vector machine, or SVM. These models aim to draw a boundary between two classes—in this case, places where fires occurred and places where they didn't. They're often used when the number of features is high but the number of samples is relatively small. In that kind of setup, SVMs can sometimes perform better than simpler models. Still, training them can take some effort. The results often depend a lot on the way the parameters are tuned, and the computing cost can grow quickly as the dataset gets bigger.

Even though traditional supervised learning models aren't as advanced as some of the newer ones, they still seem to be useful in many wildfire prediction studies—especially in cases where the datasets aren't too large or where researchers need something easier to explain. They're often a good starting point, though they might not always keep up when the number of variables grows or the relationships

between them get more complicated. That could be one reason why recent studies have started trying out other directions, like using more advanced models or mixing multiple approaches together.

To give a clearer picture of how traditional supervised learning methods have been applied in wildfire prediction, Table 2.1 presents a comparison of several representative studies. It includes information on the algorithms they used, the types of input data, the regions studied, and how the models performed. While the data sources and modeling techniques vary across studies, one common pattern seems to be that ensemble methods—such as Random Forest or Boosted Regression Trees—and optimization-based approaches like MARS-DFP often produce more accurate and stable results. Reviewing these models in a structured way helps show what they're good at, where they might fall short, and offers practical ideas that could guide model design in this study.

Table 2.1 Comparison of representative supervised learning models used in wildfire prediction

No.	Author & Year	Model Used	Input Data Types	Study Area	Performance Metrics	Notes
1	Sayad et al. (2019)	ANN, SVM	NDVI, LST, Thermal anomalies (remote sensing)	Canada	ANN: 98.32%, SVM: 97.48%	Processed on Databricks; high coverage and accuracy
2	Bui et al. (2019)	MARS + DFP	Slope, NDVI, temp., humidity, land use, etc.	Lao Cai, Vietnam	AUC: 0.95; Accuracy: 86.57%	Outperformed other traditional models
3	Same study	ANN, RBFANN, ANFIS, RF	Same as above	Same as above	AUC: 0.83–0.90	Slightly lower than MARS-DFP
4	Pourghasemi et al. (2020)	BRT, GLM, MDA	10 environmental and climatic factors	Fars Province, Iran	BRT AUC: 0.89; GLM: 0.864	BRT more robust across regions
5	Wood et al. (2021)	Custom (non-regression)	Elevation, wind, NDVI, slope, etc. (13 vars)	Montesinho, Portugal	MAE, RMSE: strong performance	Focused on burned area; works well with unbalanced data

Note: AUC = Area Under the Curve; MAE = Mean Absolute Error; NDVI = Normalized Difference Vegetation Index; DFP = Differential Flower Pollination.

2.3.2 Applications of Image Recognition and Computer Vision in Wildfire Prediction

Some researchers have started to explore the use of image-based data to support wildfire prediction. Rather than relying only on structured variables, these approaches focus on visual inputs—like satellite imagery, drone footage, or even

thermal images—to detect early signals of fire. Things such as smoke, heat signatures, or visible burn patterns can sometimes be picked up in images before a fire is officially reported. In theory, combining visual clues with environmental data might help models make more informed predictions

In earlier studies that used visual data, many researchers started with fairly simple techniques based on color. A common approach was to convert the images into a different color format—like RGB or YCbCr—and then apply a threshold rule to flag areas that might resemble smoke or fire. These methods were pretty straightforward and didn't take much computing effort, which made them attractive early on. But they also came with some trade-offs. Their accuracy could be affected by lighting, shadows, or background features, sometimes causing the system to misidentify what it was seeing.

Over time, as deep learning tools became more accessible, the focus began to shift toward more advanced models like convolutional neural networks (CNNs). These models don't rely on pre-set rules. Instead, they're trained using large image datasets, where each photo has been labeled beforehand. That allows them to learn more complicated visual patterns on their own. Some studies have used CNNs to identify heat signals in infrared imagery, while others apply them to classify burned areas using satellite photos. While CNNs seem to work better than older methods in many cases, they also require more data and stronger computing resources—both of which can be a challenge in practice.

Some studies have also looked into combining visual inputs with more traditional environmental data. One idea is to use features that come from images—such as NDVI, which gives an estimate of how healthy the vegetation is, or measurements of smoke thickness—and mix them with things like temperature, humidity, or topography. This kind of combination might help the model get a

broader view of the surrounding environment, which in theory could lead to better predictions. That said, putting all of this together isn't always straightforward. Working with visual data still comes with a few challenges of its own.

These challenges include things like poor image quality due to clouds or low light, inconsistencies across different sensors, or the difficulty of collecting enough labeled samples to train a robust model. In addition, some deep learning models used for image tasks can be hard to interpret, which might be a problem in real-world use cases where people need to understand or explain the model's predictions. Despite these issues, image-based methods offer a valuable perspective for wildfire prediction, especially in areas where satellite monitoring or real-time visual feeds are available.

2.3.3 Hybrid Modeling and Optimization Methods

As more researchers try to improve wildfire prediction accuracy, some have started exploring ways to combine different methods instead of relying on just one. This general idea—mixing models or adding optimization techniques—is often called hybrid modeling. From what I've seen in the literature, this kind of approach seems especially helpful when the data is messy or when no single model performs well on its own.

One common method is to use optimization algorithms to help machine learning models train better. Instead of manually adjusting all the settings, researchers sometimes apply tools like genetic algorithms, particle swarm optimization, or differential evolution. These techniques help the model automatically search for parameter combinations that lead to better performance. For example, when using a support vector machine or a random forest, an optimization algorithm might make it easier to tune the model and improve its accuracy.

Another direction that's been explored is combining the outputs from multiple models, something that's called ensemble modeling. The idea here is that since each model tends to pick up slightly different features or trends in the data, bringing them together—maybe by averaging their predictions or letting them vote—could lead to more reliable results. In some studies, researchers have even taken it a step further by using what's known as stacking. That's where the predictions from several base models are fed into a final model, which tries to learn how to weigh or merge them effectively. This kind of setup has been tested in a number of wildfire-related projects. In general, it seems to help in some cases, though the outcome can vary a lot depending on the data and the context.

Choosing which variables to include in a model is something that also comes up a lot in wildfire prediction. Since fire risk could be influenced by many different factors—weather, vegetation conditions, and human-related features, for instance—it's not always obvious which ones will actually help the model most. When too many variables are included, the model might end up being slower to train or, in some cases, more prone to overfitting. To manage that, some studies have used methods like principal component analysis or recursive feature elimination. These tools are designed to narrow down the input set, so the model can concentrate more on the features that matter most.

That said, putting different methods together doesn't always make things simpler. In fact, hybrid models are often more complex and might take longer to build and fine-tune. Depending on the setup, some optimization tools introduce randomness into the training process, which means the results might not always be exactly the same every time. And as the models get more layered, it becomes harder to explain how they actually reach a conclusion—which can be tricky if people need to trust or interpret the predictions.

Still, these approaches seem to offer some useful ways of handling prediction tasks that involve messy or high-dimensional data. They won't work for every situation, but they could be a good fit when simpler models fall short. In practice, it's probably less about chasing the most advanced method and more about finding a balance—between accuracy, simplicity, and how clearly the model's behavior can be understood.

Table 2.2 Comparison of Common Hybrid Modeling and Optimization Strategies

Method Type	Core Idea	Application Approach	Advantages	Limitations
Model + Optimization	Use optimization algorithms to automatically adjust model parameters (e.g., depth, learning rate)	Apply genetic algorithms, particle swarm optimization, etc., to tune SVM, RF, or neural network models	Reduces manual tuning; improves model performance; better adaptability	High computational cost for large parameter spaces; some randomness in results
Ensemble Modeling	Combine predictions from multiple models to improve stability and robustness	Use voting, averaging, or stacking to integrate outputs from several base models	More stable results; handles different data distributions better	Increased complexity; longer training time
Feature Selection	Select the most important input variables to reduce irrelevant features	Use PCA, recursive feature elimination (RFE), or genetic selection methods to filter features	Reduces dimensionality and overfitting risk; improves training efficiency	May miss useful nonlinear relationships; performance depends on feature quality
End-to-End Hybrid	Integrate multiple stages (preprocessing, modeling, optimization) into a unified pipeline	Combine data cleaning, feature selection, and modeling into an automated workflow (e.g., AutoML)	High automation; suitable for large-scale data modeling	Less transparent; limited control over internal modeling process

Note: RF = Random Forest, SVM = Support Vector Machine, PCA = Principal Component Analysis, RFE = Recursive Feature Elimination, AutoML = Automated Machine Learning.

2.4 Summary and Research Implications

In this chapter, I tried to go through the main types of methods that have been used in wildfire prediction studies. Honestly, there doesn't seem to be one single "best" model—each method has its own strengths, but also some clear limitations depending on the data and the setting.

For example, traditional supervised models like logistic regression or decision trees are still being used quite a lot, maybe because they're easy to understand and quick to apply. They seem to be especially helpful when working with relatively clean datasets or when researchers want to clearly see which variables matter. But when things get more complicated—like when there are too many interacting variables—they often don't do so well.

On the other hand, some studies have started to use visual data like satellite images or video. These give a very different type of input, sometimes helping to spot early fire signs like smoke or burned patches. Models based on image recognition (especially deep learning ones) do look promising, but they also have downsides: they need a lot of data, take more time to train, and can be hard to explain to non-technical users.

Then there are hybrid approaches. From what I've read, combining multiple models or using optimization tools (like tuning or feature selection) may lead to better results—at least in some cases. But these methods also seem more complex, and I guess they require more decisions from the researcher: which parts to combine, what parameters to tune, and how to check if it's actually working better. In some papers, the results look very strong; in others, they seem less convincing.

Overall, it's probably fair to say that wildfire prediction is a difficult task, especially when data is messy or comes from different sources. Many of the models people use today do a decent job, but there are still some problems—like low generalizability, high training cost, or lack of clarity in how the predictions are made.

For my own work, I hope to learn from these existing studies and maybe combine some of their ideas. I'll probably focus on building a model that can work with several types of features, including geographic and weather-related ones, and possibly look into optimizing its performance using tuning or filtering methods. More details on that will come in the next chapter.