Forest Fires Burned Area Monitoring with AI Machine Learning Algorithms

Xingrui Ren*

Guangdong Guangya High School, Guangzhou, China

*Corresponding author: Rx-Riley@outlook.com

Abstract. Forest fires pose significant dangers and can have devastating effects. The major consequences of large-scale forest fires include but are not limited to, the injury or death of humans and wildlife, destruction of homes and infrastructure, loss of wildlife habitats, and the release of substantial amounts of greenhouse gases (such as carbon dioxide) into the atmosphere [1]. Due to factors like climate and weather conditions, completely preventing forest fires is often impossible, making it crucial to monitor burned areas and detect large-scale fires. Traditional methods for detecting and monitoring forest fires typically rely on local sensors [2], which can be both expensive and inefficient. Our aim is to develop an accurate AI machine learning model to predict the size of the burned area in a forest fire, thereby aiding in scale monitoring. The model is built using an open-access dataset from Montesinho Natural Park in Portugal [2]. The methodologies employed in building the models include multiple regression, SVR regression, ANN regression, KNN classification, and SVM classification. By using these ML models, forest fires can be detected and monitored more efficiently in real time, potentially reducing property damage, optimizing resource allocation, and mitigating environmental harm.

Keywords: Machine learning, Multivariate regression, Multivariate classification, Artificial neural networks, Forest fire.

1. Introduction

Forest fires pose serious risks, including loss of life, destruction of property, and the release of greenhouse gases like carbon dioxide [1]. While their occurrence, particularly in dry seasons, is often unavoidable, accurate detection and monitoring are crucial. Traditional fire monitoring methods, such as deploying sensors across forests, have proven costly and inefficient [2]. With the rise of data-driven techniques, the need for accurate models to predict burned areas has become essential for improving fire monitoring efforts.

This project aims to develop machine learning models to predict the size of a forest fire's burned area using data from Montesinho Natural Park, Portugal [2]. A sensitivity analysis will identify the optimal input features for the model. The goal is to apply the trained model for real-time fire prediction and monitoring in Montesinho, with the potential for global application in forest fire monitoring.

Traditional methods like geographic information systems, satellite data, and drone mapping offer valuable insights but cannot forecast burned areas and support rapid resource allocation. To overcome this, our project applies AI techniques to predict the burned area of forest fires, enabling optimized resource deployment, such as assigning the right number of firefighters. Framed as a decision-making problem, we will integrate decision-making techniques with AI algorithms to enhance the solution.

The frequency of catastrophic wildfires is rising, causing significant property loss and environmental destruction. Data from the National Fire Protection Association shows that each wildfire damages millions of assets and threatens human lives [3]. Wildfires also harm wildlife and release large amounts of dust and carbon dioxide, impacting air and water resources. Accurate burned area prediction can help minimize property damage, making wildfire prediction critical for protecting the environment and communities.

2. Literature Review

Erten, Kurgun, and Musaoglu [4] used satellite imagery and Geographic Information Systems (GIS) to analyze forest changes caused by wildfires. They compared datasets from 1992 (pre-fire) and 1998 (post-fire) using the Maximum Likelihood supervised classification algorithm to estimate forest loss. By assigning weights to parameters like vegetation, slope, aspect, and proximity to roads and settlements, they also created a fire risk zone map. The study identified high-risk areas but did not account for seasonal variations, limiting classification accuracy. Additionally, the weighting method lacked generalizability, as each high-risk forest area requires a tailored fire risk assessment approach due to unique environmental factors.

Ozbayoglu and Recep Bozer [5] estimated fire size shortly after outbreaks using multilayer perceptron (MLP), radial basis function networks (RBFN), and support vector machines (SVM), with k-means clustering for result grouping. The MLP model performed best, achieving a 65% success rate in predicting the burned area. However, the dataset's noise and irrelevant features likely impacted the accuracy of their results.

Berlinger, Preisler, and Benoit [6] used statistical models for probabilistic forest fire risk assessment in specific regions and elevations. Their historical dataset, from Federal lands in Oregon (April 1989–December 1996), was analyzed using a 'spatial-temporal conditional intensity function.' The study provided probabilities of forest fires occurring in specific regions and months. While this supports empirical modeling for fire prediction, a limitation is the lack of real-time weather data, relying solely on historical datasets.

3. Research Methods

3.1. Input Selection and Preprocessing

The Fine Fuel Moisture Code (FFMC) measures moisture in surface litter, the Duff Moisture Code (DMC) reflects moisture in decomposed organic material, and the Drought Code (DC) represents soil moisture. The Initial Spread Index (ISI) combines wind speed and fine dead fuel moisture to estimate fire spread. These four indices form the Fire Weather Index (FWI), indicating fire intensity.

Our project aims to explore the relationship between fire indices and the burned area. We grouped months into four seasons to examine the correlation with seasonality and balance the dataset. Since over 60% of burned area data points are near zero, showing a right-skewed distribution, we applied a logarithmic transformation to approximate a Gaussian distribution for model training.

Variables with larger ranges can dominate the analysis, overshadowing those with smaller ranges. Standardizing the data corrects this imbalance, improving machine learning model performance and ensuring a more balanced interpretation by addressing range bias.

Histograms show that temperature, relative humidity, and wind speed follow Gaussian distributions, while Fire Weather Index (FWI) variables do not. This indicates that FWI variables need further preprocessing before applying models on them, such as removing irrelevant data points. The histograms reveal outliers, requiring further data cleaning. Outliers are defined as points outside 1.5 times the interquartile range (IQR). After removal, the dataset contains 190 valid inputs and outputs, ready for training regression and classification models.

Forest fire observations were classified as large or small based on burned areas. Instances with a burned area of zero were removed, and a threshold of 3.2 hectares was set based on a credible reference [2]. Fires $\geq$ 3.2 hectares were labeled as large (Label #1), while those smaller were labeled as small (Label #0). Fig. 1 shows the frequency distribution, which indicates no significant bias between the two classes despite the slight imbalance. This classification will help to prioritize high-risk areas for intervention. The entire dataset was randomly divided into a training set (80%) and a testing set (20%), and all evaluations were conducted on the testing dataset.

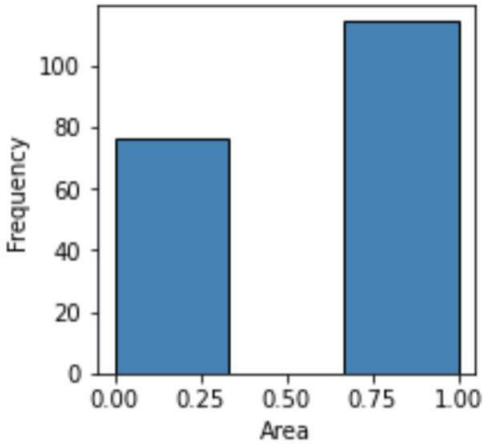

Fig. 1: Frequency Distribution Plot of Large / Small Forest Fires

3.2. Input Selection and Preprocessing

The dataset, focused on burned forest areas, contains multiple inputs and one output. This study aims to determine the relationship between these inputs and the burned area, as well as their interactions. Both conventional methods and machine learning methods will be used to uncover patterns in the data. These models can be divided into regression models and classification models for the regression of burned forest areas and the classification of burned forest area scales, respectively.

Multiple Regression: Multiple regression extends linear regression by modeling a single output with multiple inputs, adding terms for each explanatory variable [7]. While useful for evaluating multiple factors, it is less accurate for complex regression tasks.

SVR (Support Vector Regression): Support Vector Regression (SVR) performs nonlinear regression with multiple inputs and a single output, using the same model as SVM. SVR creates a margin around the function where no loss is calculated for samples within it; only those outside contribute to the loss. The model minimizes margin width and total loss, offering significantly higher accuracy than linear regression, especially for complex data relationships. [13]

Artificial Neural Network (ANN): An Artificial Neural Network (ANN) model for regression is a machine learning technique that mimics the way biological neurons process information. In regression tasks, the ANN predicts a continuous output based on input features. The network consists of layers: an input layer, one or more hidden layers, and an output layer. Each layer contains interconnected nodes that process inputs through weighted connections and activation functions. During training, the network adjusts these weights using optimization algorithms like Stochastic Gradient Descent (SGD) to minimize the error between predicted and actual values. ANN models are particularly useful for capturing complex, non-linear relationships in the data, making them effective for tasks where traditional linear models fall short. [14]

K-Nearest Neighbors (KNN): KNN is a non-parametric classification algorithm that classifies objects based on the k nearest training examples in the feature space. The output is determined by a majority vote among the k neighbors, assigning the object to the most common class. [15]

Support Vector Machine (SVM): SVM is a popular classification algorithm that performs well in high-dimensional spaces by using support vectors to form decision boundaries, making it memory-efficient [10]. Its adaptability is enhanced with kernel functions like Linear, Polynomial, and Radial Basis Function (RBF), allowing it to handle diverse data types.

Decision Tree: The Decision Tree Classifier is widely used for classification and regression tasks, creating decision rules at each split based on input-output correlations, making it simple and interpretable. [16]

3.3. Process of Machine Learning Model Development

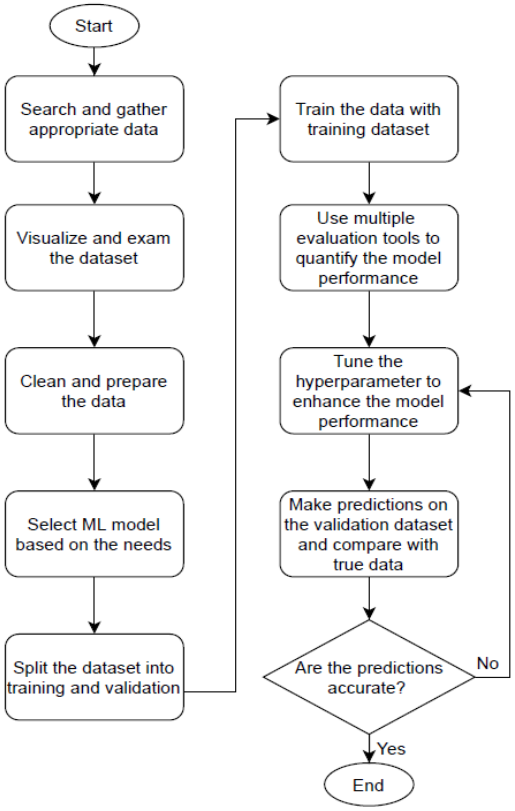

Fig. 2: Flow Chart of Machine Learning Model Development

The process of developing a machine learning model has been outlined in Fig. 2. This flowchart outlines the machine learning model development process, starting with data collection, followed by data visualization and cleaning to prepare the dataset. After selecting an appropriate model, the data is split into training and validation sets. The model is trained on training data, and its performance is evaluated using various metrics. If performance is unsatisfactory, hyperparameters are tuned to improve accuracy. The model's predictions are then tested on the validation set. If the predictions are accurate, the process concludes; if not, tuning and evaluation continue until optimal results are achieved. This iterative process ensures the best model is developed.

3.4. Process of Machine Learning Model Development

This project will use five regression performance metrics: Root Mean Square Error (RMSE), Mean Absolute Percent Error (MAPE), and Coefficient of Determination (R^2). For classification, Accuracy and Recall will also be applied. These metrics ensure a comprehensive assessment of the predictive performance of different machine learning algorithms. Formulas for each metric are provided below.

Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2}$$

Mean Absolute Percent Error (MAPE):

$$MAPE = 100 \times \frac{1}{N}\sum_{i=1}^{N} \frac{|e_i|}{|y_i|}$$

Coefficient of Determination (R-square):

$$SSE = \sum_{i=1}^{N} e_i^2$$

$$SST = \sum_{i=1}^{N} (y_i - \bar{y})^2$$

$$R^2 = 1 - \frac{SSE}{SST}$$

Accuracy (for Classification):
Accuracy = (TP + TN) / (TP + FP + TN + FN)
Recall (for Classification):
Recall for Positive Label = TP / (TP + FN)
Recall for Negative Label = TN / (TN + FP)
Where: TP is True Positives; FP is False Positives; FN is False Negatives; TN is True Negatives.

3.5. Hyperparameter Tuning

Hyperparameter tuning is essential in machine learning for optimizing algorithm performance. Below is a list of key hyperparameters and tuning methods for the algorithms used in this project:

K-Nearest Neighbors (KNN) for Classification:

Number of Neighbors: This is the primary hyperparameter, determining the number of neighbors KNN considers in classification. The tuning range is set from 1 to 20.

Algorithm: Several algorithms can be applied to compute the nearest neighbors, including 1) Ball Tree, 2) KD Tree, and 3) Brute-force Search. All will be tested and evaluated. Sklearn's KNN can also automatically select the most suitable algorithm.

Support Vector Regression (SVR):

Kernel: The kernel function transforms inputs into the necessary forms. Potential kernels include 1) Linear, 2) Polynomial, 3) Radial Basis Function (RBF), and 4) Sigmoid. Each will be tested and evaluated.

Degree: If the Polynomial kernel is selected, the degree of the polynomial function can be adjusted. The degree will be tuned between 2 and 6.

Multiple Regression:

Intercept: This hyperparameter can be set to either true or false, determining whether to include the intercept or assume the data is centered. Both options will be tested to determine which yields better accuracy.

Support Vector Machine (SVM):

Kernel Types: Three kernel types will be used for SVM: Linear, Polynomial, and RBF. Each will be tested to identify the best-performing SVM configuration.

Artificial Neural Network (ANN)

Activation Function: An activation function determines whether a neuron should be "activated" or not. It introduces non-linearity to the model, allowing it to capture complex patterns and relationships in the data. Several activation functions are evaluated including Sigmoid, Tanh, ReLU, and Leaky ReLU.

Optimizer: The optimizer is responsible for adjusting the model's weights to minimize the loss function. The optimizer updates the weights and biases of the network during training to find the optimal model parameters that lead to the lowest possible error. The optimizers evaluated are Stochastic Gradient Descent (SGD) and Adam.

Learning Rate: The learning rate is a hyperparameter in neural networks that controls how much the model's weight adjusts during each update in the training process. The learning rate was tuned from 0.01 to 0.05.

Batch Size: Batch size represents the number of training examples used in one forward and backward pass through a neural network during the training process. The batch size was tuned from 16 to 64.

4. Research Analysis

4.1. Regression

The first regression model used was multiple regression. We first checked for multicollinearity, where one explanatory variable is highly correlated with another, making it difficult to isolate individual effects. We calculated the correlation matrix and removed variables with a correlation above 0.9. Initially, regression on the non-standardized dataset yielded a low R-squared value of 0.09, indicating a poor fit. To improve the model, we standardized the dataset, which increased the R-squared to 0.15, showing the model's sensitivity to variable magnitudes.

The second regression model employed Support Vector Regression (SVR) with four kernels: Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid. After scaling the input dataset using StandardScaler, the RBF kernel produced an initial R-squared value of 0.044, indicating poor fit and prompting hyperparameter tuning. The linear kernel performed best, with regularization (C) set to 2.6 and epsilon to 0.1, improving the R-squared to 0.256, with a training loss of 0.0027 and validation loss of 0.01717.

The third model applied was an artificial neural network (ANN) due to the absence of clear patterns in scatter plots. Despite higher computational costs, we measured better regression performance. Models with fewer than three layers minimized overfitting, while deeper models overfitted the training data. To improve generalization, we applied dropout (0.2) and regularization. Using the Leaky ReLU activation function, we tested both SGD (momentum 0.1) and Adam optimizers, which showed similar results. Fine-tuning hyperparameters—learning rate (0.02), batch size (50), and regularization weight (0.01)—yielded an R-squared of 0.289, outperforming SVR and multiple regression. Though ANN offered more flexibility and fault tolerance, it required more resources and longer training times.

Table. 1 compares the R^2 values of three regression models: Multiple Regression, Support Vector Regression (SVR), and Artificial Neural Network (ANN). The R^2 value, which measures how well the model fits the data, is lowest for Multiple Regression at 0.1549, indicating a weaker fit. SVR performs better with an R^2 of 0.2565, while ANN achieves the highest R^2 value of 0.2891, suggesting that it captures the relationships in the dataset more effectively than the other two models. Overall, ANN outperforms the others in this comparison.

Table 1. Three Scheme comparing

Multiple regression	SVR	ANN
1	456	456
2	789	213
3	213	654

4.2. Classification

Classification algorithms were evaluated using accuracy and recall, the two key metrics. Recall for large fires is critical to avoid the dangerous misclassification of large fires as small, while recall for small fires minimizes incorrect classification of small fires as large. Total accuracy offers a general measure of algorithm performance.

Table. 2 compares classification algorithms based on performance metrics, showing only the best configuration for each. For SVM, the RBF kernel was used, as it achieved the highest accuracy. KNN performed best overall, with the highest total accuracy (74%), recall for small fires (60%), and solid recall for large fires (83%). Although SVM with the RBF kernel had a higher recall for large fires (91%), its low recall for small fires and accuracy made it unsuitable. Logistic Regression exhibited a

similar issue. While Decision Tree avoided this problem, its performance was weaker than KNN across all metrics.

KNN, with an optimal K value of 14 (as shown in Fig. 3), emerged as the most effective algorithm. This model is valuable for distinguishing between large and small forest fires, enabling fire departments to allocate resources more efficiently. Since most fires are small and pose minimal risk, accurate classification can help prioritize responses.

Table 2. Flow Chart of Machine Learning Model Development

	SVM - RBF Kernel	Logistic Regression	Decision Tree	KNN
Total Accuracy	63%	61%	50%	74%
Large Fire Recall	91%	83%	52%	83%
Small Fire Recall	20%	27%	47%	60%

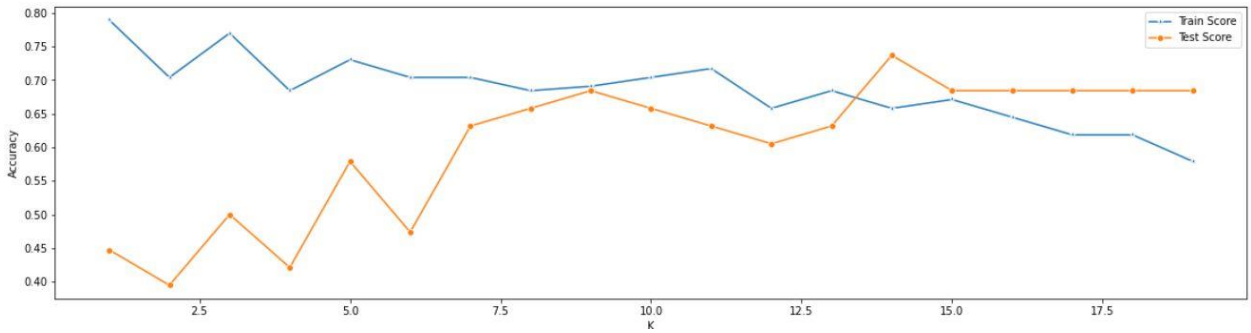

Fig. 3: Accuracy vs. K values for KNN model

5. Discussion and Conclusion

In conclusion, forest fires remain a significant issue, with over 8,700 wildfires burning more than 4.1 million acres in California this year alone [2]. While eradicating the problem is unlikely, machine learning offers valuable tools for understanding fire patterns. A major challenge in our research project was the weak correlations in the dataset, likely due to fire prevention efforts that controlled fire spread, resulting in many small burned areas. Including data on prevention efforts could improve the dataset's value. Given that the dataset contains real-world noise from meteorological sensors, achieving an R-squared value of 0.25 and a classification accuracy of 74% is a reasonable outcome.

Future work should address data quality limitations by incorporating more detailed environmental variables, such as vegetation types, soil moisture, and wind patterns, to improve model accuracy. Higher-resolution spatial data from technologies like drones or satellites could provide more granular insights into fire progression. Expanding the dataset across regions and seasons would enhance model generalization and reduce regional or seasonal biases.

Further research could explore advanced ensemble methods, such as Random Forests and Gradient Boosting, or hybrid models to better capture non-linear relationships. Transfer learning on larger, global datasets may improve generalization, and evaluating model performance across different fire-prone regions could enable more targeted fire management strategies.

References

[1] Pacific Northwest Research Station, "Fire Effects on the Environment." [Online]. Available: https://www.fs.usda.gov/pnw/page/fire-effects-environment. [Accessed: 20-Oct-2020].

[2] P. Cortez and A. Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. In J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, December, Guimaraes, Portugal, pp. 512-523, 2007. APPIA, ISBN-13 978-989-95618-0-9.

[3] Matthew Foley,"The High Cost of Wildfire in 2018", The magazine of the National Fire Protection Association. https://www.nfpa.org/News-and-Research/Publications-and-media/NFPA-Journal/2019/November-December-2019/Features/Large-Loss/Wildfire-Sidebar

[4] Erten, Esra & Kurgun, Vedat & Musaoglu, Nebiye. (2002). Forest Fire Risk Zone Mapping From Satellite Imagery And GIS: A Case Study.

[5] Özbayoğlu, A. M., Bozer. Recep. Estimation of the Burned Area in Forest Fires Using Computational Intelligence Techniques. (2012)

[6] D.R. Brillinger, H.K. Preisler, J.W. Benoit, Risk assessment: a forest fire example. In D. Goldstein, T. Speed, (Eds.), Statistics and Science: a Festschrift for Terry Speed, IMS Lecture Notes Monograph Series, vol. 40, Institute of Mathematical Statistics, Beachwood, OH, 177-196, (2003).

[7] Grant, P. (2020, March 23). Understanding Multiple Regression. Retrieved from https://towardsdatascience.com/understanding-multiple-regression-249b16bde83e

[8] Luo, S. (2018, October 14). Optimization: Loss Function Under the Hood (Part I). Retrieved from https://towardsdatascience.com/optimization-of-supervised-learning-loss-function-under-the-hood-df1791391c82

[9] Sebastian Ruder (2016, January 19) An overview of gradient descent optimization algorithms https://ruder.io/optimizing-gradient-descent/index.html#stochasticgradientdescent

[10] Sklearn.svm.SVR documentation, Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[11] [11]Keras documentation: Layer weight regularizers. Retrieved from https://keras.io/api/layers/regularizers/

[12] Sklearn.svm.SVC documentation, Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[13] Awad, M., Khanna, R. (2015). Support Vector Regression. In: Efficient Learning Machines. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-5990-9_4

[14] Zou, J., Han, Y., So, SS. (2008). Overview of Artificial Neural Networks. In: Livingstone, D.J. (eds) Artificial Neural Networks. Methods in Molecular Biology™, vol 458. Humana Press. https://doi.org/10.1007/978-1-60327-101-1_2

[15] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. (2003). KNN Model-Based Approach in Classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds) On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science, vol 2888. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39964-3_62

[16] Suthaharan, S. (2016). Decision Tree Learning. In: Machine Learning Models and Algorithms for Big Data Classification. Integrated Series in Information Systems, vol 36. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7641-3_10