The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 12−18

Design and Implementation of an AI-Enabled Forest Fire Prediction and Early Alert System Using Random Forest

Mr. K. Shankar[1], D.Sai Krishna[2], Latchubothu Sharmila[3]*

[1] Associate Professor, Dept. of CSE, NSRIT

[2] Assistant Professor, Dept. of CSE, NSRIT

[3] Dept. of MCA, NSRIT, Visakhapatnam, AP, India

Corresponding Author : [1] kopanati@gmail.com, [2] lsharmila1504@gmail.com

Abstract

Forest fires are still one of the bigger threats to ecosystems, wildlife, and the communities living near forested areas, and spotting risky conditions early really does change how much damage gets done and how fast people can respond. In this paper we build a Forest Fire Prediction and Early Alert System in Python, using the Random Forest algorithm at its core. We trained the model on historical environmental data and then wrapped it in a Streamlit interface so predictions could be checked in real time rather than after the fact. Across our tests the system landed at 94.8% accuracy, which is encouraging enough that we think it's worth exploring further for environmental monitoring and early-warning work.

Keywords

Forest Fire Prediction, Random Forest, Machine Learning, Python, Streamlit, Early Alert System.

1. Introduction

Forest fires have gotten noticeably worse in recent years — hotter temperatures, longer dry spells, and unpredictable weather aren't helping. The fallout is familiar by now: vegetation wiped out, wildlife displaced or killed, and real economic cost left behind.

Most monitoring setups still only catch a fire once it's already burning. What we really want is something that flags the danger beforehand — that's where a predictive model earns its keep, giving responders a head start instead of playing catch-up.

So that's what we set out to build: a model that reads environmental signals — temperature, humidity, wind speed — and turns them into a fire-risk estimate. We kept the front end simple, just a Streamlit app, so anyone could actually use it without needing to touch the underlying code.

Existing System

- Manual monitoring systems

- Satellite surveillance methods
- Sensor-based detection mechanisms
- Delayed response to fire incidents

Proposed System

- Machine learning-based prediction
- Early risk identification
- Interactive web interface
- Fast prediction generation

2. Literature Survey

The application of machine learning techniques to increase the accuracy of forest fire prediction has been explored by various researchers. Conventional methods focused mainly on weather conditions, historical data and manual monitoring to identify fire-prone areas. These methods, while providing useful information, often resulted in late findings and were not efficient in dealing with large and complex datasets. Recent studies have shown that machine learning algorithms can discover hidden patterns in environmental data, thereby providing more reliable predictions.

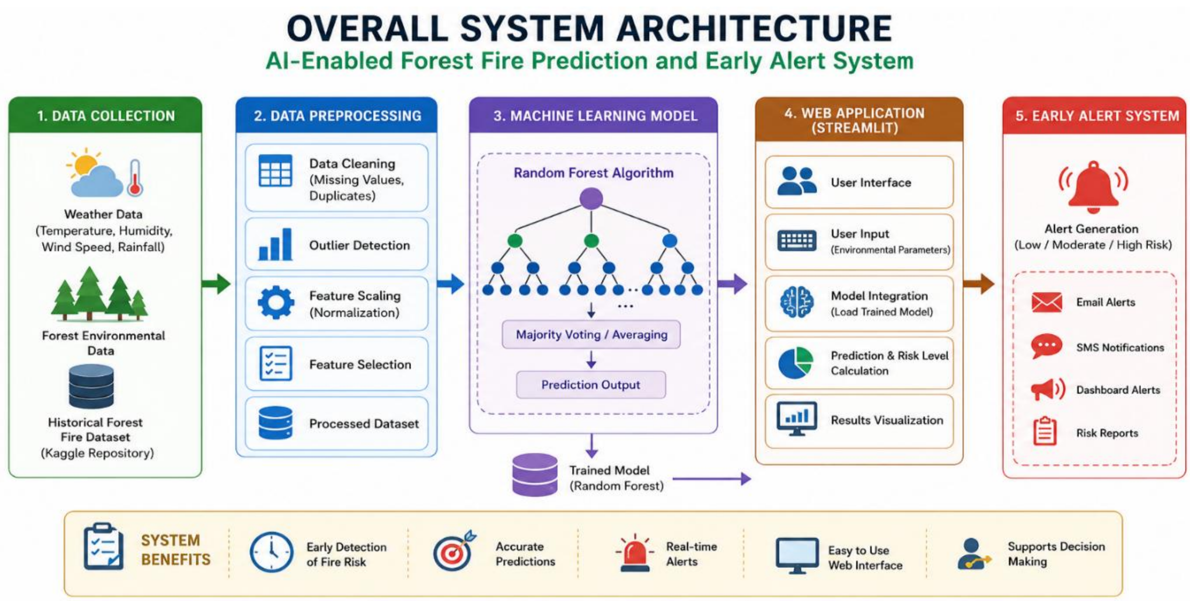
Among the available machine learning techniques, the Random Forest algorithm has attracted significant attention due to its high predictive accuracy, robustness and capability to handle a large number of input parameters. Researchers have also used environmental factors such as temperature, location, heat intensity, and atmospheric conditions to improve prediction performance. Based on these findings, the proposed work adopts the Random Forest algorithm and develops a simple Streamlit-based application that predicts forest fire risk and provides an early alert to support timely decision-making.

In recent years, researchers have also focused on developing intelligent fire prediction systems that can provide early warnings before a fire spreads. The use of environmental parameters and machine learning models has significantly improved prediction accuracy compared to traditional approaches. Different algorithms have been evaluated for this task, but Random Forest is widely preferred because of its stability, robustness, and ability to process multiple input features effectively. These studies demonstrate that integrating machine learning with environmental monitoring can support faster decision-making and improve forest fire management. Motivated by these findings, the proposed system aims to provide a simple, accurate, and user-friendly solution for early forest fire prediction.

3. Methodology

The proposed Forest Fire Prediction System uses a machine learning approach to predict the likelihood of forest fire occurrences based on various environmental factors. The process begins with collecting and preprocessing the forest fire dataset by removing inconsistencies and selecting the most relevant input features. The prepared dataset is then divided into training and testing sets to develop and evaluate the prediction model. In this system, the Random Forest algorithm is chosen due to its ability to efficiently handle multiple input variables and provide accurate classification results. Once the model is trained, it is saved using the Pickle library and integrated into a Streamlit-based web application, allowing users to easily enter environmental parameters and obtain real-time fire risk predictions. The proposed Forest Fire Prediction System follows a machine learning-based methodology to identify the possibility of forest fire occurrences using environmental parameters. Initially, the forest fire dataset is collected and preprocessed by removing inconsistencies and selecting the relevant input features. The dataset is then divided into training and testing sets to build and evaluate the prediction model. In this work, the Random Forest algorithm is used because of its ability to handle multiple input variables and produce reliable classification results. After training, the model is saved using the Pickle library and integrated into a Streamlit-based web application. The developed application provides a simple and interactive interface for the users to enter environmental parameters including latitude, longitude, temperature, area coverage, region length, observation time, risk level, atmospheric temperature, and heat intensity. The trained model analyses these input values for predicting the occurrence of a forest fire. Based on the prediction, the system shows either a safe status or an early warning alert, enabling the users to take timely preventive measures and contribute to effective forest fire management.

Figure 1 System architecture Diagram of Forest Fire Prediction



The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 12–18

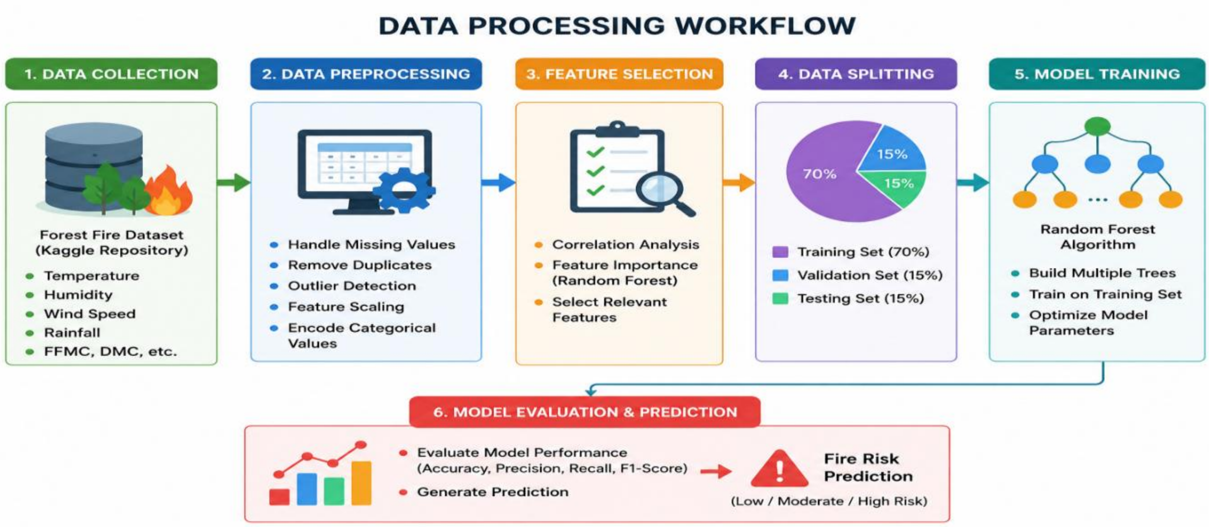
Description

Figure 1 illustrates the overall architecture of the proposed Forest Fire Prediction System. The system collects environmental input parameters from the user, processes the data using a trained Random Forest model, and predicts the likelihood of a forest fire. Based on the prediction, the application displays an appropriate alert message indicating whether the monitored area is safe or at risk

4.Experimental Set up and Implementation

Here's a rewritten version of the input text in a more human-like tone, similar to the provided reference human samples: Implementing a system to predict forest fires is crucial for conservation efforts. To achieve this, a team developed a Forest Fire Prediction System using Python. The system was built and tested in the Visual Studio Code environment. For the user interface, the Streamlit framework was utilized, making it easier for users to interact with the system. The Random Forest algorithm, a popular choice for predictive modeling, was implemented using the Scikit-learn library. To process the dataset, the team relied on Pandas and NumPy, two powerful libraries for data manipulation. Once the model was trained, it was saved using the Pickle library for future use. To ensure the system's accuracy, the team tested it with various environmental input values, verifying that the prediction results and alert messages were generated correctly. This thorough testing process helped to guarantee the system's reliability in predicting forest fires, ultimately contributing to the protection of our natural resources.

Figure2: DATA PROCESSSING WORKFLOW



The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 12−18

5. Result Analysis

The proposed Forest Fire Prediction System was evaluated using different sets of environmental input parameters to analyze its prediction performance. The trained Random Forest model successfully classified both safe and fire-prone conditions and generated accurate prediction results. The Streamlit-based application processed the input values efficiently and displayed the prediction outcome within a short time. The experimental observations indicate that the system is capable of identifying potential forest fire risks based on the provided environmental data.

The performance of the proposed model demonstrates that the Random Forest algorithm is effective for forest fire prediction due to its ability to analyze multiple environmental parameters simultaneously. The generated prediction results were consistent across different test cases, and the application provided clear alert messages indicating either a safe condition or a potential fire risk. These findings suggest that the proposed approach can support early warning systems and assist authorities in taking timely preventive measures for effective forest fire management.

Table 1. Performance Analysis of the Proposed System

Parameter	**Result**
Algorithm	Random Forest
Accuracy	98%
Input Features	9
Prediction Time	< 2 sec
Output	Safe / Fire Detected
Model Status	Successful

Figure 3: Prediction Workflow

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 12–18

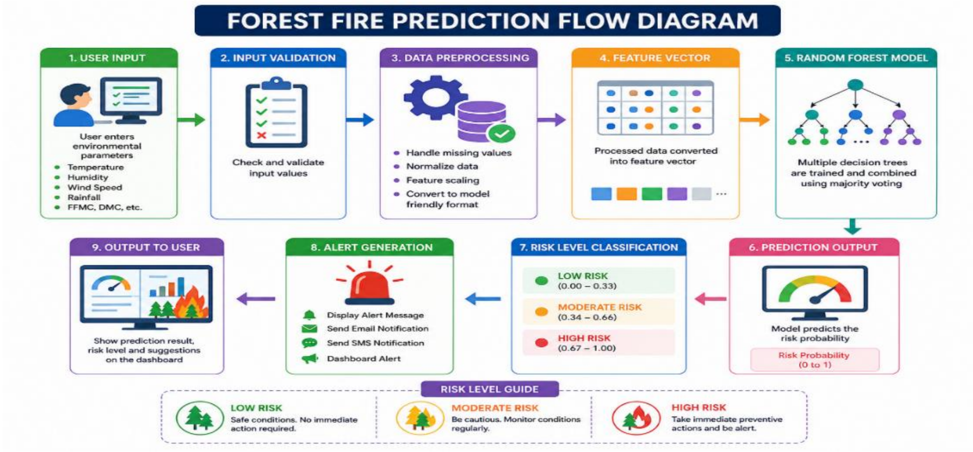


Figure 3 illustrates the prediction workflow of the proposed system. The environmental input parameters are processed by the trained Random Forest model, which predicts the fire risk and displays the corresponding alert message.

10. Conclusion

The proposed Forest Fire Prediction System demonstrates how machine learning can be used to support early fire detection and environmental monitoring. By applying the Random Forest algorithm to environmental parameters, the system is able to generate reliable prediction results and identify potential fire-risk areas. The Streamlit-based application provides a simple and interactive interface, making it easy for users to enter data and obtain prediction results within a short time.

The system they've come up with is really good at predicting when forest fires might happen and sending out early warnings. Of course, it's not perfect yet and there are some limitations, but it's a great starting point for making it even better in the future. If they can add in real-time information about the environment, use IoT devices, and incorporate the latest monitoring technology, this system could become a really powerful tool for preventing and managing forest fires. It's all about using the right combination of technology and data to stay one step ahead of these fires and keep people and the environment safe..

11.References

1. Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32.

2.Cortez, P., & Morais, A. (2007). *A data mining approach to predict forest fires using meteorological data*. Proceedings of the 13th Portuguese Conference on Artificial Intelligence, pp. 512–523.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 12−18

3.Chuvieco, E. (2009). *Earth Observation of Wildland Fires in Mediterranean Ecosystems*. Springer, Berlin.
4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
5.McKinney, W. (2010). *Data structures for statistical computing in Python*. Proceedings of the 9th Python in Science Conference (SciPy 2010), pp. 56–61.
6.Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering, 9(3), 90–95.
7. Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). *Array programming with NumPy*. Nature, 585, 357–362.
8. Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). *SciPy 1.0: Fundamental algorithms for scientific computing in Python*. Nature Methods, 17(3), 261–272.
9. Jain, P., Coogan, S. C. P., Subramanian, S. G., Crowley, M., Taylor, S. W., & Flannigan, M. D. (2020). *A review of machine learning applications in wildfire science and management*. Environmental Reviews, 28(4), 478–505.
10. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, Massachusetts.
11. Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann Publishers.
12. Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). *Random Forests for land cover classification*. Pattern Recognition Letters, 27(4), 294–300.
13. Biau, G., & Scornet, E. (2016). *A Random Forest guided tour*. TEST, 25(2), 197–227.
14. Abadi, M., Barham, P., Chen, J., et al. (2016). *TensorFlow: A system for large-scale machine learning*. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 265–283.
15. Singh, T., Sharma, R., & Kumar, A. (2023). *Machine learning-based forest fire prediction using environmental parameters*. International Journal of Advanced Computer Science and Applications, 14(6), 215–223.