MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888

AN AI AND IOT-BASED EARLY WILDFIRE RISK PREDICTION SYSTEM

Ms. Nithya V
Assistant Professor, Department of Computer Science and Engineering,
KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India
nithyavaj@gmail.com

Harsha R M
Department of Computer Science and Engineering,
KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India
rmharsha355@gmail.com

Janasri R
Department of Computer Science and Engineering,
KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India
janasrisri209@gmail.com

Dharanika S
Department of Computer Science and Engineering,
KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India
dharanikas2005@gmail.com

Abhay Krishna U K
Department of Computer Science and Engineering,
KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India
abhaykrish04@gmail.com

Abstract— Wildfires are among the most serious environmental challenges we face today. Their damage does not end once the flames are extinguished. Forests remain scarred, animals lose their habitats, air quality worsens, and nearby communities continue to feel the effects long after the fire has vanished. Although fire monitoring technology has made progress over the years, most systems still only react after smoke or flames are visible. By that point, the fire often spreads quickly, and the chance to stop it early is mostly gone. This challenge inspired the work presented here: an early wildfire risk prediction system that uses AI and IoT to identify dangerous conditions before a fire starts. The system's concept is both simple and practical. Low- cost sensors connected to an ESP32 microcontroller continuously monitor temperature, humidity, and smoke levels. Instead of relying on fixed limits that struggle to keep pace with changing environmental conditions, the system learns what normal behavior looks like. When it detects something unusual, it treats it as an early warning sign. When these unusual patterns are identified, the system classifies the situation into three clear states: Safe, Warning, or Alert. This information is shown on a live dashboard, making it easy to understand and respond to in real-time. Test results show that this method produces fewer false alarms than traditional systems while delivering warnings earlier and more reliably. Because the system is affordable, easy to scale, and requires minimal computing power, it is especially suitable for remote forest areas, where early action can prevent significant damage.

Keywords—Wildfire Risk Prediction, Internet of Things (IoT), Environmental Monitoring, Anomaly Detection, Isolation Forest, Machine Learning, ESP32

I. INTRODUCTION

Over the past few decades, wildfires have become both more frequent and more destructive. Rising temperatures, longer dry seasons, climate change, and increasing human activity near forests have all contributed to this trend. What starts as a small fire can quickly escalate into a large-scale disaster, affecting nature, property, and people across vast regions[7].Wildfires do more than burn trees. They destroy entire ecosystems, reduce plant and animal life, pollute the air, and create serious health risks from prolonged smoke exposure. The economic impact is also significant, resulting in major losses to agriculture, buildings, and local economies[15].Despite ongoing advancements in technology, most wildfire detection methods still respond too late. Tools like satellite imagery, surveillance cameras, lookout towers, and basic sensor systems typically detect fires only once they have already started. Satellite systems can experience delays or be obstructed by cloud cover, cameras depend on clear visibility, and basic sensors often struggle in changing weather conditions. These limitations can lead to delayed warnings, false alerts, or fires being completely overlooked[25].The growth of the Internet of Things (IoT) has made continuous and low-cost environmental monitoring possible. Sensors can now gather data on temperature, humidity, and gas levels from multiple locations simultaneously. However, gathering data alone is not enough. To prevent fires early, that data must be analyzed quickly and accurately. This is where artificial intelligence becomes crucial. AI can learn normal environmental patterns and detect minor changes that could signal an increased fire risk .With this goal in mind, this work focuses on predicting wildfire risk early instead of simply detecting fires after they start. The proposed system uses low-cost sensors and an ESP32 microcontroller to collect real-time data. This information is analyzed to identify unusual behavior, and the results are clearly shown on a dashboard as Safe, Warning, or Alert. The aim is to provide a practical, affordable, and timely early warning system that helps reduce wildfire damage[18].

II. LITERATURE SURVEY

A. IoT-Based Wildfire Monitoring Systems

Many researchers have studied how IoT and wireless sensor networks can be used to monitor forests and detect wildfires[7]. These systems typically place sensors throughout forest areas to measure temperature, humidity, and gas levels. The collected data is sent to a server or cloud platform to generate alerts. While these methods have improved response times compared to manual monitoring, most still focus on detecting fires only after they have begun[25].Chan et al. reviewed several IoT-based wildfire detection systems and demonstrated that environmental sensing is effective for forest monitoring. However, they also highlighted that most systems rely on fixed limits or visible fire indicators such as smoke and flames. As a result, these systems remain reactive and offer limited support for early risk prediction[29].

B. Cloud and Communication-Based Fire Detection Approaches

To manage large amounts of data and enhance system performance, some researchers have turned to cloud-based designs. Ioannidis et

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888

al. developed a forest fire monitoring system using cloud services and IoT sensors. While this approach improved flexibility and data sharing, it still focused on detecting fires after they ignite rather than predicting risk in advance[7].Other studies looked into long-range communication technologies like LoRa for forest monitoring. Apriani et al. created a LoRa-based sensor network that functioned over long distances while consuming very little power. However, it continued to rely on fixed thresholds, making it vulnerable to weather changes and more likely to trigger false alarms[29].

C. Machine Learning-Based Wildfire Detection Techniques

Machine learning has also been used to enhance wildfire detection. Zhao et al. proposed a system that leveraged sensor data and preset limits to trigger fire alerts. Although this reduced detection delays, it could not predict fire risk before ignition[4]. Dampage et al. applied supervised learning methods to classify fire and non-fire conditions using historical data[10]. While these methods achieved high accuracy, they required large labeled datasets and significant computing resources. This makes them difficult to deploy on low-power sensor devices and limits their use across different regions.

III. PROBLEM STATEMENT

Although many wildfire monitoring systems are available today, most still operate reactively, detecting fires only after they have started. These systems depend on simple indicators like smoke, flames, or high temperature. Consequently, warnings often arrive too late, allowing fires to spread quickly and cause serious environmental and economic damage. Fixed thresholds also struggle to adjust to changing weather conditions, leading to false alarms or missed detections. Many current solutions face additional practical challenges. Some machine learning models require labeled fire data, which is hard to obtain. Others need more computing power than small IoT devices can provide. Additionally, many systems lack the necessary integration between sensors, data analysis, and real-time visualization tools needed for quick decision-making. Together, these issues emphasize the need for a simple, flexible, and intelligent wildfire risk prediction system. Such a system should continuously monitor environmental conditions and provide early warnings before a fire starts. The proposed AI- and IoT-based solution meets this need by combining low-cost sensors with smart anomaly detection, enabling early, reliable, and practical wildfire risk warnings.

IV. PROPOSED SYSTEM OVERVIEW

The system architecture consists of three main layers:

- IoT Sensor Node Layer – ESP32 microcontroller with DHT11 and MQ-2 sensors collects real-time environmental data.
- Communication Layer – Uses MQTT protocol over Wi- Fi to transmit JSON-formatted sensor data to a local Python server.
- AI Processing & Dashboard Layer – An Isolation Forest model processes the data to detect anomalies, and a web- based dashboard displays the results.

Each IoT node operates autonomously, capturing temperature (°C), humidity (%), and smoke concentration (ppm). The ESP32 publishes the data to the MQTT broker at fixed intervals (every 10–15 seconds). The Python application subscribes to the topic, preprocesses the incoming data, and passes it through the trained machine learning model for classification.

If the system detects abnormal environmental variations, it issues alerts as:

- SAFE: Normal conditions
- WARNING: Pre-ignition anomalies (temperature rise and humidity drop)
- ALERT: Confirmed smoke detection indicating possible ignition

This architecture ensures a predictive, responsive, and easily scalable system for early wildfire management.

V. METHODOLOGY

This system follows a step-by-step process that combines IoT sensors and machine learning to detect possible forest fire risks. It continuously collects environmental data, sends it wirelessly, cleans and processes it, extracts meaningful patterns, and uses an AI model to identify unusual conditions. Based on the model's output, the system generates alerts in real time. Overall, this workflow helps recognize early environmental changes that may lead to wildfire incidents.

Data Acquisition:

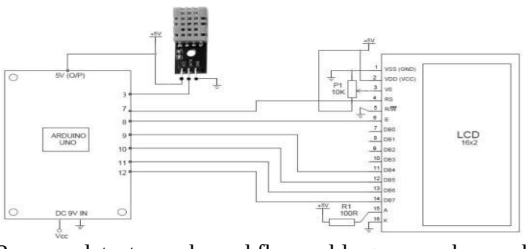
A key part of the system is its ability to collect real-time environmental data using an IoT sensor network deployed in the target area. These sensors monitor critical factors such as temperature, humidity, and smoke levels, which play an important role in assessing fire risk.

Hardware Components:

ESP32 Microcontroller: The ESP32 acts as the main controller in the system. It connects with the sensors, reads their data, processes the information, and transmits it wirelessly. Its built-in Wi-Fi and Bluetooth features make it efficient and suitable for IoT-based applications.

DHT11 Sensor: The DHT11 sensor measures temperature and relative humidity. While it is a low-cost sensor, it provides sufficient accuracy for basic wildfire monitoring and environmental tracking.

Figure: 1.1 : DHT11 Sens



MQ-2 Gas Sensor: The MQ-2 sensor detects smoke and flammable gases such as carbon monoxide and methane. Its output reflects smoke concentration, helping the system detect possible fire or smoke events.

Data Sampling and Encoding:

The ESP32 collects sensor readings every 10 seconds to ensure continuous monitoring. Each reading is tagged with a timestamp and stored in JSON format, which makes the data easier to transmit, store, and analyse.

Data Transmission:

The system uses the MQTT protocol to send sensor data wirelessly. MQTT is lightweight, fast, and well-suited for IoT environments with limited computing resources Architecture Setup:

Publisher: ESP32 device sending sensor data

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888

Broker: Mosquitto MQTT server (local or cloud-based) Subscriber**:** Python-based AI module that processes incoming data
This follows a publish–subscribe communication model, ensuring reliable and low-latency data transfer even when network connectivity is unstable.
Communication Flow:

The ESP32 first connects to a Wi-Fi network using login credentials. After connecting, it publishes sensor data to the backend through an MQTT topic. The Python subscriber listens to this topic and stores incoming data in a database for further processing and prediction.
Advantages of MQTT Transmission:

- Uses minimal network bandwidth
- Supports Quality of Service (QoS) for reliable delivery
- Enables multi-device communication
- Ensures fast and responsive data transmission

During testing, the system recorded an average data delay of less than 200 milliseconds, making it suitable for real- time fire alerts.

a. Data Preprocessing:
Since raw sensor data may contain noise, missing values, or minor errors, preprocessing is required to improve accuracy before training the AI model.

b. Data Cleaning:
Missing readings are estimated using linear interpolation or the last recorded valid value, ensuring smooth data flow.

c. Outlier Removal:
Unrealistic values (e.g., temperature above 60°C or humidity below 5%) are removed to prevent incorrect model learning.

d. Normalization:
Sensor values are scaled between 0 and 1 using Min– Max normalization, ensuring fair contribution from each feature and faster model learning.

$$z = \frac{v - v_{low}}{v_{high} - v_{low}}$$

where:

- z is the normalized feature value
- v represents the original sensor reading
- v_{low} is the minimum observed value of the feature
- v_{high} is the maximum observed value of the feature

This step ensures uniform feature contribution and faster model convergence.

e. Sensor Calibration:
The MQ-2 sensor may fluctuate due to environmental conditions such as humidity. To improve accuracy:

- A baseline smoke level is recorded for 15 minutes
- Future readings are compared against this baseline
- Only meaningful smoke deviations trigger alerts This helps reduce false alarms.

$$D = R - B$$

where:

- D is the calibrated (adjusted) smoke value
- R represents the current raw smoke sensor reading
- B denotes the baseline smoke level

f. Noise Reduction:
A Moving Average Filter is applied:

$$p = \frac{1}{W} \sum_{u=0}^{W-1} r_{t-u}$$

where:

- p represents the smoothed value
- r_{t-u} denotes the raw sensor readings over the window
- W is the number of samples used for averaging This smooths rapid fluctuations, maintaining trend

consistency while preserving anomaly sensitivity.

g. Feature Extraction:
Tracks how quickly temperature changes over time. A steady rise may indicate abnormal heating and early fire risk.
Temperature Trend (ΔT/Δt):
Measures rate of temperature change per unit time.

$$q = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (v_k - \bar{v})^2}$$

where:

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888

- q denotes the temperature variation measure
- v_k represents individual temperature readings
- $\bar{v}$ is the mean temperature value
- M is the total number of temperature samples

Dryness Index (DI):

Combines temperature and humidity to estimate environmental dryness. Higher values indicate greater fire risk.

$$R = \frac{P}{Q}$$

where:

- R is the Dryness Index
- P represents the temperature reading
- Q represents the humidity reading

Higher DI values indicate higher fire risk. Feature Vector Composition

For each time window t, the input vector is represented as:

$V_I = [p_i, q_i, r_i, \Delta p_i, \Delta q_i, R]$

Where:

- p_i is the temperature value
- q_i is the humidity value
- r_i is the smoke sensor reading
- Δp_i represents the rate of change of temperature
- Δq_i represents the rate of change of humidity
- R is the Dryness Index

These engineered features enable the model to detect subtle anomalies that static threshold-based systems might overlook. Visualization can illustrate temporal variation of features.

- Rising temperature curve
- Falling humidity trend
- Increasing dryness index approaching alert threshold Such patterns visually validate the pre-ignition phase.

h. Model Development:

The system uses unsupervised machine learning, meaning it learns what normal environmental behaviour looks like and flags unusual conditions — even without labelled fire data.

1. Isolation Forest Algorithm:

The Isolation Forest (iForest) algorithm isolates anomalies by recursively partitioning data using random splits.

- The temperature change rate is calculated as:

$$\Delta p_i = \frac{p_i - p_{i-1}}{i - (i-1)}$$

where:

- Δp_i represents the temperature change rate
- p_i is the current temperature reading
- p_{i-1} is the previous temperature reading

2. Humidity Trend (ΔH/Δt):

Indicates decreasing moisture level — a precursor to ignition:

$$\Delta q_i = \frac{q_i - q_{i-1}}{i - (i-1)}$$

where:

- Δq_i represents the rate of change of humidity
- q_i is the current humidity reading
- q_{i-1} is the previous humidity reading

3. Gas Concentration Stability (σ Smoke):

Evaluates the short-term standard deviation of smoke readings to detect volatility.

Key Concepts:

- Normal instances require more partitions to isolate (longer path length).
- Anomalies are isolated quickly (shorter path length). The anomaly score A(u)for a data point x is calculated as

$$A(u) = 2^{\frac{L(u)}{K(m)}}$$

where:

- A(u) denotes the anomaly score of the sample
- L(u) represents the average path length of the sample across all isolation trees
- K(m) is the normalization factor based on the dataset size mmm

The normalization factor was calculated as:

$$K(m) = 2H(m-1) - \frac{2(m-1)}{m}$$

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888

where:

- H(m−1) is the harmonic number of (m−1)
- m is the total number of data samples

As the anomaly score A(u) approached 1, the corresponding data point was considered highly anomalous, indicating an increased wildfire risk.

i. Model Training

- Dataset: 2,000 records under normal conditions.
- Model: 100 trees, contamination factor = 0.05.
- Framework: scikit-learn implementation (Isolation Forest class).
- Training Time: <2 seconds on standard laptop (Intel i5, 8GB RAM).

j. Comparative Models

To validate performance, additional models were evaluated:

- Autoencoder: Deep neural model minimizing reconstruction error.
- SVM (Support Vector Machine): Classifier distinguishing normal vs abnormal samples.
- Random Forest: Ensemble classifier used as a supervised baseline.

Isolation Forest outperformed others in accuracy, simplicity, and speed, making it optimal for resource- constrained IoT systems.

k. Model Evaluation Metrics

Model efficiency was measured using:

- Precision (A): $\frac{X}{X+Y}$
- Recall (R): $\frac{X}{X+Z}$
- Latency: Average prediction time per sample.
- F1-score: $2 \times AxB/A+B$

The Isolation Forest achieved an F1-score of 0.94 and an inference latency of 25 ms.

l. Prediction and Alert

Once trained, the model is integrated with the live sensor stream to provide real-time anomaly prediction and automated alerting.

m. Continuous Inference

The Python-based subscriber reads MQTT data, preprocesses it, extracts features, and sends it to the trained model for scoring. Each instance receives an anomaly score between 0 and 1.

n. Alert Thresholds

Decision logic converts anomaly scores into interpretable alert states:

Score Range	Status	Description
0.0–0.4	SAFE	Normal environmental conditions
0.4–0.6	WARNING	Pre-ignition anomaly detected
>0.6	ALERT	Critical fire risk; smoke likely detected

TABLE I. ALERT THRESHOLDS

A secondary condition validates alerts — if the MQ-2 smoke reading exceeds the baseline, the alert level escalates automatically.

Alert Workflow

- Anomaly Detected: System classifies reading as WARNING or ALERT.
- Event Logging: Timestamp and sensor data are recorded in a database.
- Dashboard Update: Real-time color change and message display.
- Notification: Optional triggers (buzzer, email, or mobile app alert).

VI. IMPLEMENTATION

The proposed system uses an integrated hardware and software framework. It brings together environmental sensing, edge computing, wireless communication, machine learning analysis, and real-time visualization. The system has four main parts: sensor node hardware, embedded firmware, backend analytics, and dashboard visualization.

A. Hardware Implementation

The sensing unit is built using an ESP32 microcontroller, selected for its low power consumption, built-in Wi-Fi capability, and sufficient processing resources for real-time sensor acquisition. To capture critical environmental data for fire risk assessment, the ESP32 is interfaced with two types of environmental sensors

DHT Temperature and Humidity Sensor:

This sensor continuously measures ambient temperature and relative humidity, which are key indicators of fire-prone conditions. Low humidity and rising temperatures significantly increase the risk of wildfires.

MQ-2 Smoke and Gas Sensor:

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888

The MQ-2 sensor detects the presence of smoke and combustible gases. Although the proposed system focuses on pre-ignition risk, gas concentration trends provide early
indications of abnormal environmental behavior.
All sensors are powered directly from the ESP32 and connected through digital and analog input pins, creating a unified sensor node. This node is designed to be compact, low-cost, and suitable for deployment in remote forest locations.

B. Embedded Firmware Implementation

The ESP32 firmware is developed using Arduino IDE / VS Code (Platform IO) and is responsible for real-time data acquisition and transmission.
The firmware performs the following operations: Sensor Initialization:

At startup, the ESP32 initializes the DHT and MQ-2 sensors and establishes a Wi-Fi connection using predefined network credentials.
Periodic Data Acquisition:

Sensor values for temperature, humidity, and gas concentration are sampled at fixed intervals (e.g., every 5– 10 seconds). Multiple readings are averaged to reduce noise and improve stability.
Data Formatting:

The collected sensor readings are formatted into a structured JSON packet containing sensor values, timestamp, and device ID.
Wireless Data Transmission:

The ESP32 sends the JSON data to the backend server using HTTP REST API requests over Wi-Fi. This ensures reliable and lightweight communication suitable for IoT environments.
Backend and Machine Learning Implementation:

The backend server is implemented using Python and Flask, acting as an interface between the sensor nodes and the AI prediction module.
Data Reception and Storage:

The Flask server receives incoming sensor data through REST API endpoints and stores it in a local database or CSV file for analysis and historical tracking.
Data Preprocessing:

Incoming data is normalized and cleaned to handle noise or missing values before being passed to the machine learning model.
AI-Based Risk Prediction:

An Isolation Forest anomaly detection model is used to analyze environmental behavior. The model is trained using historical "normal" environmental data. During runtime, the model evaluates incoming sensor readings and assigns an anomaly score that reflects wildfire risk.
Risk Classification:

Based on anomaly scores, environmental conditions are categorized into:
SAFE: Normal environmental behavior WARNING: Deviations indicating increasing fire risk
ALERT: High-risk conditions requiring immediate attention
This approach allows early risk assessment without waiting for actual fire ignition.

C. Dashboard and Visualization

A web-based dashboard is developed using HTML, CSS, and JavaScript to provide real-time system monitoring. Live visualization of temperature, humidity, and smoke levels. Display of predicted risk status (SAFE / WARNING/ ALERT). Historical trend graphs for environmental parameters. Automatic alert notifications when ALERT conditions are detected. The dashboard communicates with the backend server through REST APIs, ensuring real-time updates and user-friendly interaction.

VII. CONCLUSION

The performance of the proposed system was evaluated using real-time sensor data collected from ESP32-based IoT nodes under varying environmental conditions. The Isolation Forest anomaly detection model was trained using normal environmental data to establish baseline behavior for temperature, humidity, and smoke parameters. During evaluation, the system consistently identified abnormal environmental patterns characterized by gradual temperature increases combined with declining humidity levels, even in the absence of smoke. These conditions resulted in elevated anomaly scores, leading to the generation of early Warning alerts that indicate pre-ignition wildfire risk. When smoke was introduced alongside anomalous environmental patterns, the system reliably escalated the alert level, demonstrating its ability to identify high-confidence fire events. The model exhibited stable performance during continuous real-time data streaming, with low computational overhead and minimal processing latency, making it suitable for real-time deployment. The multi-level alert mechanism enhanced situational awareness by distinguishing between normal, early-risk, and high-risk conditions, enabling timely intervention. Overall, the experimental observations confirm that integrating unsupervised machine learning with IoT-based sensing enables effective early wildfire forecasting by detecting subtle environmental deviations prior to ignition, while maintaining robust and reliable system behavior.

VIII. REFERENCES

[1] R. Singh et al., "Active wildfire detection via satellite imagery and machine learning," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 18, pp. 1–15, 2025.
[2] B. Thies et al., "Machine-learning wildfire susceptibility mapping for Germany (2003–2023)," Natural Hazards, vol. 119, no. 2, pp. 987–1008, 2025.
[3] S. Grunewald et al., "Modelling current and future forest fire susceptibility in Brandenburg," International Journal of Wildland Fire, vol. 34, no. 1, pp. 45–62, 2025.
[4] Y. Zhao et al., "Near-real-time wildfire progression mapping with VIIRS," Remote Sensing of Environment, vol. 311, pp. 113985, 2025.
[5] L. Z. Zhang et al., "Near-real-time wildfire detection with Himawari-8/9 using MSSTF," IEEE Transactions on Geoscience and Remote Sensing, vol. 63, pp. 1–14, 2025.
[6] Y. Yu et al., "Deep learning and remote sensing image analysis for wildfire prediction: A review," Remote Sensing, vol. 17, no. 3,

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888

pp. 412–439, 2025.
[7] M. Ejaz et al., “A machine learning pipeline survey for wildfire risk prediction,” Environmental Modelling & Software, vol. 177, pp. 105656, 2025.
[8] J. Zhang et al., “Forest fire prediction framework based on multiple machine learning models,” Applied Soft Computing, vol. 149, pp. 110957, 2025.
[9] R. Abohaia et al., “A machine learning framework for predicting wildfire dynamics across Australia’s seven regions,” Ecological Informatics, vol. 79, pp. 102261, 2025.
[10] M. Ejaz et al., “Comprehensive survey of the machine learning pipeline for wildfire risk prediction,” Information Fusion, vol. 99, pp. 101897, 2025.
[11] Y. Xu et al., “Deep learning for wildfire risk prediction: Integrating remote sensing data,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 206,
pp. 280–296, 2025.
[12] N. Thakur et al., “Wildfire susceptibility mapping using GIS and machine learning,” Geocarto International, vol. 40, no. 1, pp. 1–21, 2025.
[13] F. Andrianarivony et al., “Machine learning and deep learning for wildfire spread prediction: A systematic review,” Fire, vol. 7, no. 2, pp. 58–81, 2024.
[14] M. Moghim and A. Mehrabi, “Wildfire assessment using machine learning algorithms in different regions,” Natural Hazards, vol. 116, no. 3, pp. 2319–2341, 2024.
[15] A. Aydin et al., “Integrated risk mapping for forest fire management: A case study of Antalya,” Sustainability, vol. 16, no. 5, pp. 2147, 2024.
[16] H. Özdemir et al., “Using machine learning to predict forest-fire probability in Mediterranean Türkiye,” Environmental Monitoring and Assessment, vol. 196, no. 2, pp. 112, 2024.
[17] S. Reddy et al., “Early fire detection using satellite time-series data and machine learning,” Remote Sensing Applications: Society and Environment, vol. 33, pp. 101050, 2024.
[18] E. Shams Eddin, R. Roscher, and J. Gall, “Location- aware adaptive normalization: A deep learning approach for wildfire danger forecasting,” IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 9, pp. 5872–5884, 2023.
[19] J. Zhang et al., “Real-time wildfire detection algorithm based on VIIRS and Himawari-8,” Remote Sensing, vol. 15, no. 14, pp. 3569, 2023.
[20] S. Sathishkumar et al., “Forest fire and smoke detection using deep learning-based approaches,” Expert Systems with Applications, vol. 223, pp. 119929, 2023.
[21] J. Zhang et al., “A weighted contextual active fire detection algorithm,” International Journal of Applied Earth Observation and Geoinformation, vol. 119, pp. 103260, 2023.
[22] Y. Kang et al., “A deep learning model using geostationary satellite data for forest fire detection with reduced detection latency,” Remote Sensing of Environment, vol. 278, pp. 113071, 2022.
[23] Y. Zhao and Y. Ban, “GOES-R time series for early detection of wildfires with deep GRU network,” Remote Sensing of Environment, vol. 270, pp. 112856, 2022.
[24] A. Rafaqat et al., “Remote sensing drivers of Pakistan wildfires and occurrence probability,” Natural Hazards, vol. 112, no. 1, pp. 457–478, 2022.
[25] A. Bot et al., “Systematic review of machine learning for wildfire decision support,” Fire Safety Journal, vol. 131, pp. 103608, 2022.
[26] B. Seydi et al., “Fire-Net: A deep learning framework for active forest fire detection,” IEEE Access, vol. 10, pp. 121345–121358, 2022.
[27] L. Ding et al., “Wildfire detection via deep learning on geostationary imagery,” IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–13, 2022.
[28] S. Kondylatos et al., “Wildfire danger prediction and understanding with deep learning,” Remote Sensing, vol. 14, no. 6, pp. 1438, 2022.
[29] J. Phelps et al., “Comparing calibrated statistical and machine learning methods for wildland fire occurrence prediction,” International Journal of Wildland Fire, vol. 30, no. 3, pp. 207–220, 2021.
[30] P. Jain et al., “A review of machine learning applications in wildfire science and management,” Environmental Modelling & Software, vol. 125, pp. 104606, 2020.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 35 Issue 2, 2025, Pages: 2881-2888