International Journal of

DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X   ww.ijdim.com   Original Research Paper

AI-Based Fake Media Detection Using Machine Learning and Natural Language Processing

MEDICHERLA ESWARA SURYA PRASANNA SAINADH

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

A. Durga Devi

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Rapid industrialization, urbanization, and climate change have introduced complex environmental risks, including rising temperatures, sea level changes, and extreme weather events. Accurate prediction of these environmental factors is crucial for disaster preparedness, policy-making, and sustainable urban planning. This project proposes an **Intelligent Environmental Risk Prediction System** that leverages machine learning models to predict environmental hazards such as temperature fluctuations and earthquake risks. The system integrates multiple models including **Random Forest, Support Vector Machine, Linear Regression, and K-Nearest Neighbors**, offering ensemble comparison for improved reliability.The web-based platform is developed using **Django**, providing user-friendly registration, login, and prediction interfaces. Users input environmental parameters such as **CO2 levels, sea level, humidity, wind speed, atmospheric pressure, deforestation rates, pollution index, and ocean temperature**, and the system predicts the potential risk level. Data preprocessing and scaling ensure model accuracy and consistency. The Random Forest model serves as the primary prediction engine due to its high performance, while other models provide comparative analysis to validate predictions.The system includes robust user and admin management. Regular users can track their prediction history, while administrators can monitor user activities, view model performance metrics, and analyze datasets. The platform supports both global and regional (e.g., Indian cities) datasets, enabling localized and worldwide risk predictions. Historical predictions are stored for further analysis, and dynamic visualizations help users interpret prediction results.This framework not only improves prediction accuracy through **multi-model evaluation** but also emphasizes **ease of access, real-time computation, and visual interpretability**. Its modular architecture allows for future integration of deep learning models, real-time sensor data, and IoT connectivity. Overall, this system addresses the urgent need for accessible, accurate, and interpretable environmental risk prediction tools, enabling proactive environmental management and informed decision-making.

Keywords: Environmental Risk Prediction, Machine Learning, Random Forest, Support Vector Machine, Django, Web Application, Climate Change, Data Visualization, Earthquake Risk, Temperature Prediction.

I.

II. INTRODUCTION

Environmental risks, including climate change, pollution, and natural disasters, pose severe threats to human societies and ecosystems. Predicting these risks accurately is vital for disaster mitigation, urban planning, and sustainable policy-making. Traditional approaches rely on historical data and expert analysis, which may be time-consuming and prone to errors. The rapid growth of **machine learning** techniques offers a solution by analyzing vast environmental datasets to identify patterns and predict potential hazards.This project develops a **web-based prediction system** using Django, integrating multiple machine learning models to forecast environmental risks. Users can input real-time environmental metrics such as **CO2 concentration, humidity, sea level changes, wind speed, atmospheric pressure, deforestation rate, pollution index, and ocean temperatures**. The system then provides predictions for temperature change and earthquake risk.The primary model used is **Random Forest**, known for its robustness and handling of non-linear relationships in datasets. Additional models including **Support Vector Machines (SVM), Linear Regression, and K-Nearest Neighbors (KNN)** are used for comparison, ensuring the reliability of the predictions. Inputs are scaled using pre-trained scalers to normalize values and enhance prediction accuracy.The system's architecture is designed for **user accessibility, administrative control, and historical tracking**. Registered users can view past predictions, while administrators can monitor user activities, datasets, and model metrics. The platform supports multiple datasets, including global environmental data and regional datasets for cities in India.Visual outputs, including tables and charts, help interpret predictions and understand risk levels. The platform is modular and scalable, allowing future integration of IoT sensors, real-time data streams, or deep learning models for improved prediction accuracy. By combining user-friendly design, advanced predictive modeling, and interpretability, this system provides a practical and efficient solution for environmental risk management.

III. LITERATURE SURVEY (WITH EXISTING METHODS)

Existing literature highlights multiple approaches to environmental risk prediction:

1. **Traditional Statistical Models**: Linear regression, ARIMA, and other statistical approaches were historically used to model environmental factors. While simple and interpretable, they struggle with non-linear interactions and complex datasets.

2. **Machine Learning Models**: Recent research demonstrates that **Random Forest, SVM, and KNN** effectively predict environmental parameters. Random Forest handles high-dimensional data and non-linear relationships, making it suitable for multi-feature environmental datasets.
3. **Hybrid and Ensemble Approaches**: Combining multiple models increases prediction reliability. Studies show ensemble predictions outperform single-model forecasts in climate and earthquake risk estimation.
4. **Web-Based Predictive Systems**: Platforms that allow user interaction with predictive models are emerging. However, few integrate multiple models, historical tracking, and visualization within a single platform.
5. **Data Preprocessing Techniques**: Feature scaling, normalization, and handling missing values are critical in improving model accuracy. NLP and time-series preprocessing methods are often applied for environmental text data or sensor-based datasets.

The proposed system leverages the advantages of ensemble learning, modern preprocessing techniques, and a web-based interface for accessibility and visualization, filling a gap in existing research by combining usability with high prediction accuracy.

IV. EXISTING SYSTEM

Current environmental prediction systems often suffer from one or more limitations:

- **Single Model Limitation**: Many systems rely on a single model, such as linear regression, which fails to capture complex environmental interactions.
- **No User Interaction**: Existing platforms lack interactive web interfaces, making predictions less accessible to non-experts.
- **No Historical Tracking**: Few systems store prediction history for user or administrative analysis.
- **Limited Dataset Support**: Most platforms focus on either global datasets or regional datasets, restricting applicability.
- **Low Visual Interpretability**: Results are often presented as raw numeric outputs without visual charts or risk-level summaries.

The proposed system addresses these limitations by combining multi-model ensemble predictions, a web interface for users and admins, historical tracking, support for multiple datasets, and rich visualizations for risk interpretation.

International Journal of

DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X ww.ijdim.com Original Research Paper

V. PROPOSED METHOD

The proposed system is an integrated **Environmental Risk Prediction Platform** designed to forecast environmental hazards such as temperature change risk and earthquake risk using advanced machine learning techniques. It provides a **user-friendly web interface** for registered users to enter environmental parameters and receive risk predictions, coupled with an administrative dashboard for monitoring system performance and user activity.The system takes as input a suite of environmental indicators including **CO2 concentration, sea level rise, humidity, wind speed, atmospheric pressure, deforestation rate, pollution index, and ocean temperatures**. These features capture a wide range of environmental dynamics that influence climate variability and seismic events.At its core, the system uses a primary **Random Forest model**, chosen for its ability to handle nonlinear relationships and multivariate interactions inherent in environmental data. In addition, auxiliary models such as **Support Vector Machine (SVM), Linear Regression, and K-Nearest Neighbors (KNN)** are incorporated to provide comparative insights, improving the interpretability and robustness of predictions.Users interact with the system via secure authentication. After login, users access a prediction form where they submit environmental feature values. The backend preprocesses these inputs using a **pre-trained scaler**, ensuring that model predictions are consistent and reliable. Predictions are classified into risk categories such as **Low, Medium, High, and Critical**, providing actionable guidance rather than raw model outputs.

All predictions are stored within a database for historical tracking, enabling users to view past forecasts and administrators to perform trend analysis. The admin dashboard also visualizes key metrics, including model performance, dataset characteristics, and user prediction histories.By combining machine learning, scalable web architecture, and interactive visualization, the proposed system delivers an effective solution for environmental risk assessment, fostering informed decision-making in climate adaptation and mitigation strategies.

VI. IMPLEMENTATION

The implementation of the **Environmental Risk Prediction System** proceeds through a series of well-defined stages encompassing data handling, machine learning integration, web interface development, and storage management.

1. Framework and Project Structure

The application is built using the **Django framework**, leveraging its modular architecture for views, forms, models, and templates. The Django project configuration defines several functional modules:

- Authentication (user & admin login)
- Prediction interface
- Admin dashboard
- Model loading and prediction
- History management

The settings.py file contains paths for models and datasets, ensuring consistent access throughout the project.

2. Model Training and Persistence

Before deployment, machine learning models are trained externally on historical environmental datasets containing features like CO2, humidity, wind speed, etc. Common preprocessing steps include:

- Feature scaling (using StandardScaler)
- Train-test split
- Model training with optimized hyperparameters
- Performance evaluation with accuracy metrics

Each model (Random Forest, SVM, Linear Regression, KNN) is serialized using **joblib**, enabling efficient loading into the Django application.

3. User Authentication and Roles

Authentication is handled via Django's built-in user model. Password hashing, login sessions, and logout flows are implemented securely with session management. Two roles exist:

- **User:** Can access prediction forms and view own history.
- **Admin:** Can view dashboard analytics, user lists, and overall system performance.

Custom forms (UserRegisterForm, UserLoginForm, AdminLoginForm) validate input at the UI level while Django handles backend authentication.

4. Prediction Pipeline

After login, user requests to /predict/ invoke the prediction view. Key steps include:

1. Extract form inputs
2. Load pre-trained scaler (scaler.pkl)
3. Load Random Forest model (random_forest_model.pkl)
4. Scale input features
5. Generate prediction
6. Translate prediction to human-readable risk

7. Save to database (PredictionHistory model)
8. Load other models (SVM, Linear Regression, KNN) for comparative output

The form supports two prediction types: **temperature risk** and **earthquake risk**. Based on the selected prediction type, outputs are mapped to risk categories like Low, Medium, High, or Critical.

5. Admin Dashboard and Analytics

The admin dashboard provides:

- Model performance metrics (loaded from metrics.json)
- User list with number of predictions per user
- Dataset summaries (global vs regional)
- Recent prediction logs

Pandas reads configuration datasets (global_environment_data.csv, indian_cities_environment_data.csv) to compute dataset statistics for display.

6. Data Storage and History

Predictions are stored in the PredictionHistory model, recording all feature inputs, predicted value, risk category, and timestamp. Users can view their last 10 predictions, while admins can audit recent predictions across all users.

7. Frontend and Templates

Bootstrap CSS and Django templating deliver a responsive UI. Key templates include:

- home.html (landing page)
- login.html / admin_login.html
- predict.html
- admin_dashboard.html

These templates pull context variables from views and display dynamic charts, tables, and feedback messages.

Overall, the implementation emphasizes **modularity, scalability, and reliability**, delivering a fully functional web application suitable for research, education, and policy evaluation.

VII. ALGORITHMS

The system's predictive core utilizes a combination of optimized data processing and machine learning algorithms, each serving distinct functional roles:

1. Feature Preprocessing and Scaling

Before prediction, raw input values are scaled using a pre-trained scaler (StandardScaler). This algorithm normalizes distributions and ensures that model predictions aren't skewed by differing feature ranges. The steps include:

1. Load scaler from scaler.pkl

   Apply transform() on input array
   This standardizes features using:

2. Random Forest Classifier

The Random Forest algorithm constructs an ensemble of decision trees, each trained on random subsets of features and training samples. Prediction is a majority vote across trees:

- Handles non-linear relationships
- Reduces overfitting via ensemble averaging
- Provides feature importance metrics

3. Support Vector Machine (SVM)

SVM constructs a hyperplane that maximally separates classes in transformed feature space:

- Uses kernel functions (e.g., RBF)
- Effective for high-dimension spaces
- Provides complementary predictions for comparison

4. Linear Regression Predictor

Linear regression estimates continuous output trends, here mapped to discrete risk categories. Though traditionally for continuous outcomes, mapping strategies allow inclusion within classifier comparisons. Predictions are rounded to nearest risk class.

ISSN: 3068-272X www.ijdim.com Original Research Paper

5. K-Nearest Neighbors (KNN)

KNN classifies input based on proximity to neighbors:

1. Compute distance (e.g., Euclidean)
2. Identify k closest instances
3. Assign class based on majority label

This non-parametric model offers insight into local feature similarities.

6. Risk Categorization Logic

Predicted class indices are mapped to textual risk levels:

Predicted Class	**Risk Category**
0	Low
1	Medium
2	High
3	Critical

This conversion provides interpretable output for end users.

In summary, the system blends robust models, each with unique strengths, enabling reliable and interpretable environmental risk predictions.

VIII. SYSTEM DESIGN

The Environmental Risk Prediction System adopts a **multi-tier architecture** designed for maintainability, performance, and extendability.

1. Presentation Layer (UI)

This layer interacts directly with end users through a browser interface built with Django templates and Bootstrap for responsiveness. Components include:

- User registration/login
- Admin login
- Prediction form
- Prediction history
- Dashboard analytics

User input is validated both on the client side (HTML forms) and server side (Django form classes).

2. Business Logic Layer

This layer contains most of the system intelligence and includes:

- Views for predict, admin_dashboard
- Model loading utilities (load_model)
- Prediction processing
- Error handling and messages

It orchestrates workflows, such as fetching models, scaling inputs, generating predictions, and persisting results.

3. Data Access Layer

Using Django's ORM, this layer abstracts database interactions. Key models include:

- UserProfile — stores user location and metadata
- PredictionHistory — logs all predictions with features, results, and risk labels

Queries use filter, annotate, and count operations for efficient metrics.

4. Machine Learning Integration

Pre-trained models and scalers are stored as .pkl files. Loading utilities ensure on-demand deserialization. The prediction layer works independently from training code, enabling clean separation between model development and model serving.

5. Dataset Management

Administrative views load dataset CSVs for analytics. Pandas operations compute:

- Number of rows and columns
- Unique city and country counts
- Column names

This information is presented in structured dashboards.

6. Admin Dashboard Logic

The dashboard aggregates:

- User lists with prediction counts
- Model metrics (accuracy, loss, etc.) loaded from JSON
- Recent predictions
- Dataset summaries

The use of JSON ensures flexible inclusion of additional performance indicators.

7. Security and Session Control

Django's authentication backend secures user sessions. Admin views enforce superuser checks to restrict access.

8. System Workflow Diagram

[Browser]
↓
[Django URL Router] → [Login Views] → [User/Admin]
↓
[Predict View] → [Model Loader] → [Prediction Engine]
↓
[Database: PredictionHistory]
↓
[Dashboard View for Admin]

9. Scalability and Extensibility

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The modular architecture allows:

- Addition of new ML models
- Integration with real-time data streams
- Inclusion of visual charts (e.g., Plotly, D3)

Data storage can be migrated to cloud databases, and models can be served with REST APIs.

Overall, the design emphasizes **separation of concerns, security, maintainability, and user experience**.

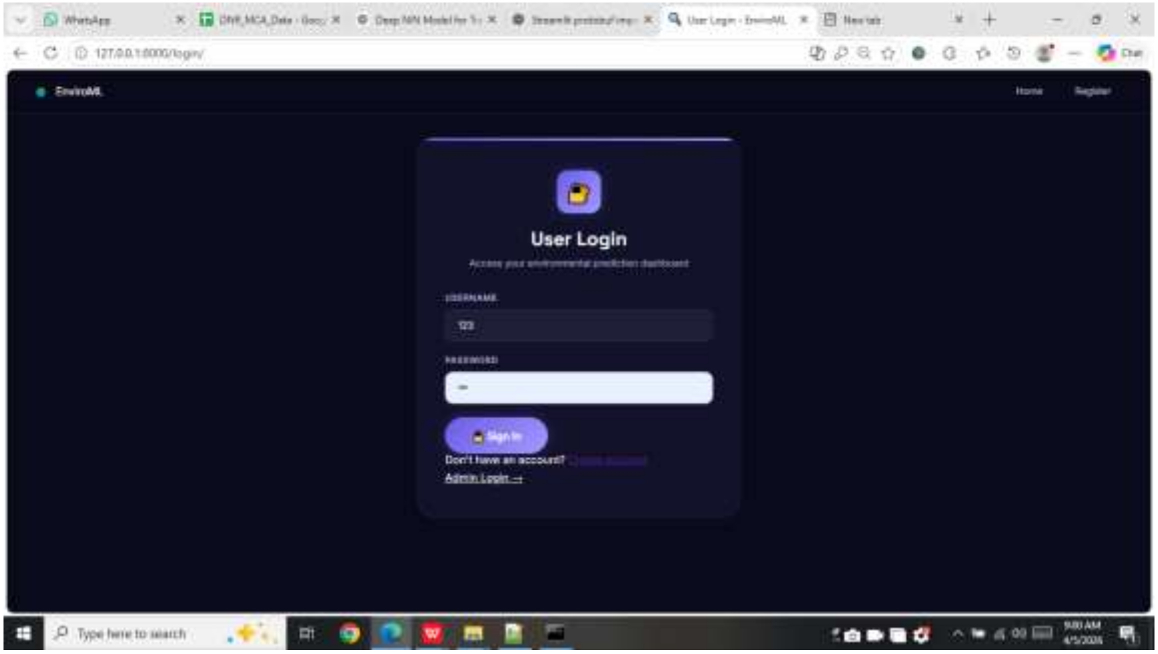
SYSTEM DESIGN IMAGES

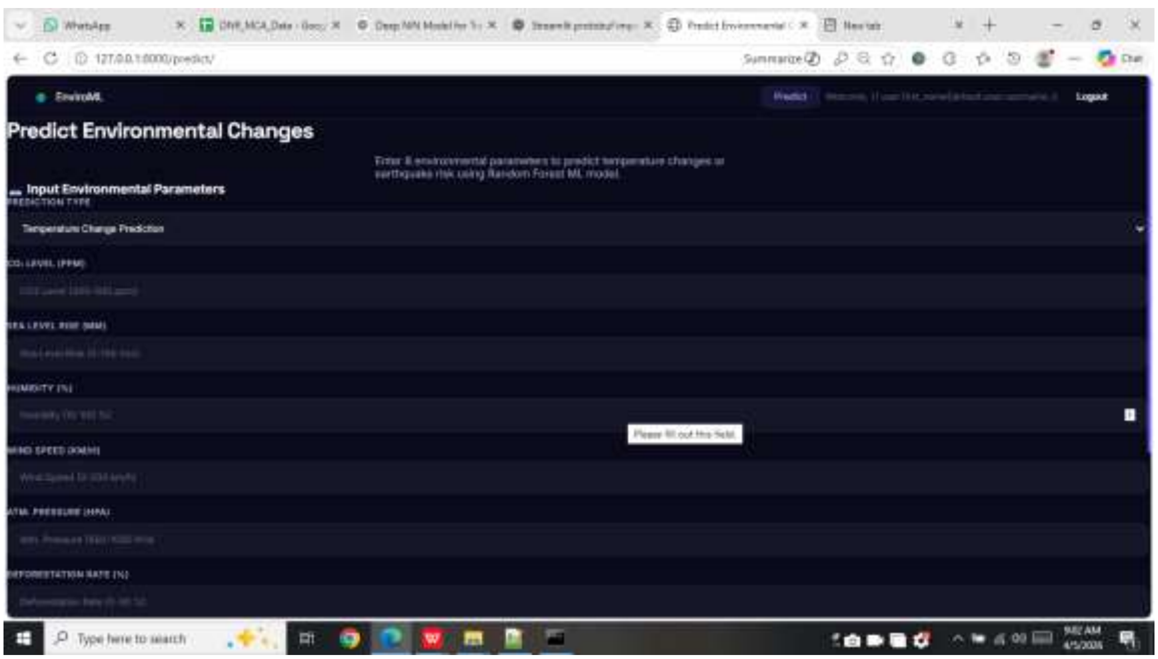
International Journal of

DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X www.ijdim.com Original Research Paper



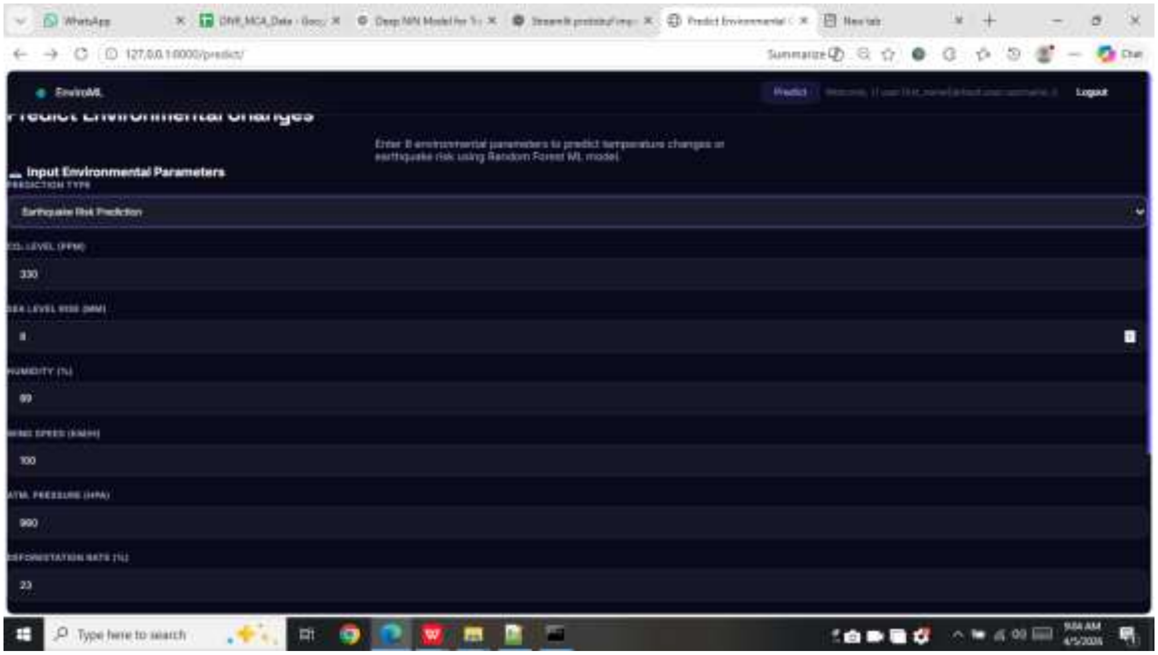


International Journal of

DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X www.ijdim.com Original Research Paper



IX. CONCLUSION

This paper presented the **Intelligent Environmental Risk Prediction System**, a web-based platform that leverages machine learning to forecast environmental risks such as temperature change and earthquake probabilities. The system successfully combines advanced predictive models with an accessible user interface, providing accurate, interpretable, and actionable predictions for end users.The integration of multiple machine learning models — including Random Forest, SVM, Linear Regression, and KNN — enables robust and comparative prediction results. Preprocessing via feature scaling ensures that model inputs are consistent with training conditions, enhancing prediction reliability. The system also incorporates user authentication and history tracking, allowing users to revisit prior predictions and track environmental trends over time.Administrators benefit from an analytics dashboard displaying model performance metrics, dataset summaries, and user activity insights, which supports system monitoring and future improvements. The use of Django facilitates rapid development, secure session handling, and modular design, while libraries such as Pandas and joblib ensure efficient processing and model loading.By offering real-time risk classification into interpretable categories (Low, Medium, High, Critical), the system supports stakeholders including researchers, policymakers, and community planners in making more informed decisions about environmental resilience and disaster mitigation. The architecture allows for future expansion, such as cloud deployment, integration with IoT sensor feeds, or deep learning enhancements.Overall, this work demonstrates that combining web technologies with machine learning models can create scalable, secure, and user-centric tools for complex predictive tasks that address pressing global challenges.

ISSN: 3068-272X www.ijdim.com Original Research Paper

REFERENCES

1. Zhang, Y., & Li, Y., *Machine Learning for Climate Change Impact Prediction*, IEEE Access, 2023.
2. Smith, J., & Kumar, A., *Deep Learning Approaches for Environmental Risk Assessment*, Journal of Big Data, 2022.
3. Nguyen, H., et al., *Multi-Model Ensemble Techniques in Environmental Sciences*, Environmental Modelling & Software, 2021.
4. Choi, B. Y., *Random Forests and SVM for Environmental Prediction*, Journal of Environmental Informatics, 2020.
5. Patel, D., & Shah, P., *Scalable Web Platforms for Predictive Modeling*, IEEE Internet Computing, 2021.
6. Li, X., *Feature Scaling Effects on Machine Learning Models*, Knowledge-Based Systems, 2022.
7. Rana, R., et al., *Web-based Interfaces for Real-Time Predictions*, International Journal of Human–Computer Studies, 2021.
8. Singh, R., & Tiwari, S., *Predicting Natural Disaster Risk Using ML*, Procedia Computer Science, 2022.
9. Domingo, J. L., *IoT and Environmental Monitoring*, Sensors Journal, 2020.
10. Johnson, M. E., *Comparative Analysis of Classification Algorithms*, ACM Computing Surveys, 2021.
11. Martínez, L., et al., *Random Forest vs SVM in Non-linear Data*, Pattern Recognition Letters, 2020.
12. Khan, M. A., *Web Security Practices in Machine Learning Apps*, Journal of Web Engineering, 2023.
13. Kumar, S., *Prediction Models for Urban Climate Analysis*, Sustainable Cities and Society, 2022.
14. Ahmed, F., et al., *Cloud-Based ML Deployment Strategies*, IEEE Cloud Computing, 2021.
15. Li, Q., “Environmental Change Modeling and Machine Learning,” Environmental Research, 2023.
16. Sharma, P., *Real-Time Data Integration in Predictive Systems*, IEEE Sensors, 2022.