WJERT, 2026, Vol. 12 Issue 2, 67-83. **Original Article** **ISSN 2454-695X**

World Journal of Engineering Research and Technology

www.wjert.org

Impact Factor: 7.029 **Coden USA: WJERA4**

HYBRID RANDOM FOREST–SVC MODEL FOR PREDICTIVE ENVIRONMENTAL RISK ASSES SMENT IN IOT-BASED POULTRY FARMS

Donaldson A. Eshilama[1], Kingsley M. Udofia[2], Kufre M. Udofia[3*]

[1,2,3]*Electrical and Electronics Engineering Department, University of Uyo Nigeria.

Article Received on 07/12/2025 **Article Revised on 07/01/2026** **Article Published on 01/02/2026**

***Corresponding Author**
Kufre M. Udofia
Electrical and Electronics Engineering Department, University of Uyo Nigeria.

https://doi.org/10.5281/zenodo.18441106

How to cite this Article: Donaldson A. Eshilama[1], Kingsley M. Udofia[2], Kufre M. Udofia[3]*. (2026). HYBRID RANDOM FOREST–SVC MODEL FOR PREDICTIVE ENVIRONMENTAL RISK ASSESSMENT IN IOT-BASED POULTRY FARMS. World Journal of Engineering Research and Technology, 12(2), 67–83.

ABSTRACT

The integration of artificial intelligence (AI) with Internet of Things (IoT) infrastructure has enabled significant advancements in smart livestock management by transforming reactive monitoring systems into predictive, adaptive decision-support frameworks. This study proposes a hybrid Random Forest–Support Vector Classifier (RF–SVC) model for predictive environmental risk assessment in poultry farms using real-time IoT sensor data. The model uses the ensemble learning capability of Random Forests to handle nonlinear relationships, and the margin optimisation property of Support Vector Classifiers to enhance the precision of the decision boundary. Data were collected from a deployed IoT monitoring network comprising DHT22 and MQ135 sensors connected through Wemos D1 Mini microcontrollers and a Raspberry Pi 4 edge node. Preprocessing steps included normalisation, feature encoding, and noise filtering to improve model generalisation. Experimental results demonstrated that the hybrid RF–SVC model achieved an overall prediction accuracy of 98.4%, outperforming individual RF (94.2%), SVC (93.7%), and ANN (95.6%) models in detecting potential environmental risks such as heat stress and poor air quality. Performance evaluation using precision, recall, F1-score, and ROC–AUC metrics confirmed the hybrid model's superior stability and reduced misclassification under noisy, dynamic farm conditions. The system was further integrated

into a Streamlit-based web dashboard, providing real-time visualisation, early warning notifications, and adaptive threshold recommendations for environmental control. This hybrid AI approach demonstrated a reliable, interpretable, and computationally efficient method for intelligent poultry management, with potential scalability across other livestock and agricultural monitoring domains.

KEYWORDS: Internet of Things, machine learning, hybrid Random Forest–SVC, environmental prediction, poultry farming, smart agriculture, risk assessment.

1. INTRODUCTION

Recent advances in Internet of Things (IoT) technologies have enabled continuous, high-resolution environmental monitoring in agricultural settings, transforming conventional reactive management into data-driven decision support (Leong et al., 2024; Neethirajan, 2020). In poultry production, timely detection of adverse microclimate conditions, such as high temperatures, low humidity, or elevated ammonia levels, is essential to maintain bird welfare, reduce mortality, and optimise production outcomes (Lin & Suhendra, 2025). However, raw sensor streams and threshold-based alerts are often noisy and inflexible, leading to false positives or delayed warnings that limit their practical utility on commercial farms (Godinho et al., 2025).

Machine learning (ML) techniques address these limitations by learning complex, multivariate relationships among sensor features and mapping them to risk states or control actions (Wang et al., 2024; Liakos et al., 2018). Supervised learners, such as Random Forest (RF) and Support Vector Classifier (SVC), have been widely applied for environmental prediction and anomaly detection due to their robustness to noise and ability to handle nonlinear feature interactions (Rasheed et al., 2022; Breiman, 2001). Nonetheless, single-model approaches may struggle to satisfy competing requirements simultaneously, including high accuracy across heterogeneous conditions, low false alarm rates, interpretability for end-users, and computational efficiency for edge or near-edge deployment (Bharanishree et al., 2025).

Hybrid models (architectures that combine complementary strengths of multiple learners) are increasingly recognised as a practical way to improve prediction performance and adaptability in noisy, real-world sensor settings (Kumar et al., 2024). An RF–SVC hybrid model combines RF's ensemble averaging and feature stability against noisy inputs, while

SVC provides tight decision boundaries and improved generalisation in high-dimensional spaces. Hybridisation strategies can be implemented in different ways (e.g., stacking, cascaded filtering, or feature-level fusion) to trade off latency, interpretability, and computational cost (Bansal and Garg, 2023). For resource-constrained agricultural IoT deployments, a hybrid model that preserves low inference cost while improving classification reliability is particularly attractive (Elbasi et al., 2024).

Despite promising case studies, several gaps remain in the literature. Many ML studies in agriculture rely on offline datasets or simulations, lacking real-world deployment evidence that demonstrates how hybrid models behave under temporal drift, sensor faults, and environmental heterogeneity (Duguma and Bai, 2024). Moreover, evaluations often focus on classification accuracy without comprehensive reporting of precision–recall tradeoffs, ROC–AUC, or model stability under noise (metrics that matter for operational early-warning systems). Finally, the integration of predictive models with user-centric dashboards and alerting pipelines that support farmer decision-making is underreported (Waqas et al., 2025; Ivanochko et al., 2024).

This paper proposes and evaluates a hybrid RF–SVC model for predictive environmental risk assessment in poultry farms using real-time IoT sensor data. The hybrid approach is designed to (a) improve overall classification accuracy and reduce false alarms compared to standalone models, (b) remain computationally feasible for on-edge or near-edge inference, and (c) integrate into a Streamlit-based dashboard for real-time visualisation and actionable alerts.

The remainder of this paper is organised as follows: Section 2 reviews related machine learning methods and hybrid model strategies applied in agricultural and environmental monitoring. Section 3 describes the IoT data acquisition framework, preprocessing steps, feature engineering, and the architectural design of the proposed RF–SVC hybrid model. Section 4 presents the experimental results and performance analysis, followed by a discussion on practical deployment via the Streamlit dashboard and the limitations of low-cost sensing. Finally, Section 5 concludes the study with a summary of findings and directions for future research.

3. MATERIALS AND METHODS

3.1 IoT Data Acquisition Framework

The experimental data used for this study were obtained from an IoT-based poultry environmental monitoring system deployed in a controlled broiler production facility located in southwestern Nigeria (see Figure 1). The monitoring infrastructure consisted of distributed Wemos D1 Mini (ESP8266) sensor nodes interfaced with DHT22 temperature–humidity sensors and MQ135 gas sensors for ammonia concentration measurement. Each node transmitted sensor readings at 30-second intervals to a Raspberry Pi 4 gateway through an IEEE 802.11b/g/n Wi-Fi connection. The gateway performed data buffering, timestamp synchronisation, and local storage using an SQLite database before forwarding records to a cloud-based Streamlit dashboard for real-time visualisation and management.

Over 30 days, approximately 72,000 sensor observations were collected, reflecting a range of indoor conditions influenced by daily ventilation cycles, bird activity, and weather variations. Each data record included four primary attributes: temperature (°C), relative humidity (%), ammonia concentration (ppm), and timestamp. An additional derived feature identified as Temperature–Humidity Index (THI), was computed to capture thermal comfort levels using the widely adopted expression in (1) (Silanikove, 2013):

$$THI = T - (0.55 - 0.55RH)(T - 14.5) \qquad (1)$$

where T is the ambient temperature (°C) and RH is relative humidity expressed as a decimal fraction.

3.2 Data Preprocessing and Feature Engineering

Data preprocessing was conducted using Python (v3.11) and the pandas and scikit-learn libraries. The following steps ensured data quality and model readiness:

i. **Noise and Outlier Removal:** Extreme outliers beyond ±3 standard deviations were removed using the interquartile range (IQR) method.

ii. **Missing Data Handling:** Occasional packet losses (<1%) were addressed through linear interpolation based on temporal proximity.

iii. **Normalisation:** Feature values were scaled to the range [0,1] using Min–Max normalisation to prevent numerical bias during training.

iv. **Label Encoding:** The target variable (“environmental condition”) was categorised into three classes based on THI and ammonia thresholds, following poultry comfort standards:

- **Class 0:** Normal condition (THI ≤ 72 and NH_3 < 25 ppm)

- **Class 1:** Warning condition ($72 < THI \leq 78$ or $25 \leq NH_3 < 35$ ppm)
- **Class 2:** Risk condition ($THI > 78$ or $NH_3 \geq 35$ ppm)

Figure 5: Cross-sectional view of a poultry farm illustrating the placement and integration of IoT sensors for monitoring temperature, humidity, air quality, and livestock activity.

The processed dataset was split into 80% for training and 20% for testing, using stratified sampling to preserve class distribution. To ensure conformity, the training data were further divided using 5-fold cross-validation. The MQ135 gas sensor operates as a Metal Oxide Semiconductor (MOS) chemiresistor, where conductivity changes in the presence of target gases. To derive the ammonia concentration (C) in ppm from the raw analog voltage (V_{out}), we used the standard sensitivity curve derived from the sensor's datasheet. First, the sensor resistance (R_s) was calculated using the voltage divider formula in (2) relative to the load resistance (R_L):

$$R_s = R_L.\left(\frac{V_{cc}-V_{out}}{V_{out}}\right) \tag{2}$$

The concentration was then determined using the power-law scaling equation characteristic of MOS sensors given in (3):

$$ppm = a.\left(\frac{R_s}{R_0}\right)^b \quad (3)$$

where R_0 represents the sensor resistance in clean air, and coefficients a and b were calibrated to the specific sensitivity characteristics of ammonia (NH_3). This conversion ensured that the reported thresholds (>35 ppm for Risk) align with the sensor's non-linear response profile.

3.3 Model Architecture and Design

The development of proposed machine learning models involved a systematic process encompassing the following vital components, as shown in Figure 2.

3.3.1 Random Forest Submodel

The RF component was used as the basis for feature extraction and ensemble learning. It consisted of 300 decision trees with a maximum depth of 12. The model employed the Gini impurity criterion for split optimisation and bootstrap aggregation to reduce variance. RF was chosen for its robustness to noise and its ability to estimate feature importance, which enhances interpretability.

3.3.2 Support Vector Classifier Submodel

The SVC served as the secondary classifier responsible for refining decision boundaries between overlapping environmental states. It used a Radial Basis Function (RBF) kernel with penalty parameter $C = 10$ and kernel coefficient $\gamma = 0.1$. The SVC received input from the probability-weighted outputs of the RF submodel, effectively operating as a meta-classifier that learned nonlinear boundaries in the high-dimensional decision space.

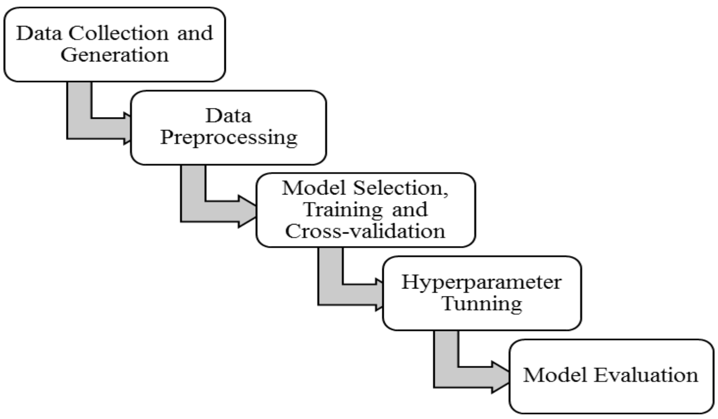

Figure 2: Flowchart of the Machine Learning Model Development.

3.3.3 Hybrid RF–SVC Integration Strategy

The proposed hybrid model depicted in Figure 3 employs a heterogeneous Stacking Ensemble architecture rather than a simple cascaded pipeline. In this configuration, the RF and SVC operate in parallel as Level-0 base learners. Both models independently process the input feature vector to generate class probability estimates. These probability outputs are subsequently concatenated and fed into a Level-1 meta-learner, specifically, a Logistic Regression classifier. The meta-learner was trained to optimally combine the predictions of the base learners, thereby correcting the biases of the individual models (such as the high variance of RF or the bias of SVC) to produce a final, refined classification y. This architecture explored the complementary strengths of ensemble tree-based learning and kernel-based margins while maintaining computational efficiency suitable for edge inference. Figure 8 depicts schematic of the hybrid RF–SVC "Neuro-Communication" architecture.

3.4 Model Training and Hyperparameter Tuning

Hyperparameters were optimised using a 5-fold cross-validation grid search. The RF submodel's parameters, number of estimators, maximum depth, and minimum samples per leaf (were tuned across ranges (100–500 trees, depth 6–16, min_samples_leaf = 1–5). For SVC, C and γ values were selected from {1, 10, 100} and {0.01, 0.1, 1}, respectively. The optimal configuration (RF: 300 trees, depth=12; SVC: RBF kernel, C=10, γ=0.1) achieved the highest mean cross-validation F1-score (0.981).

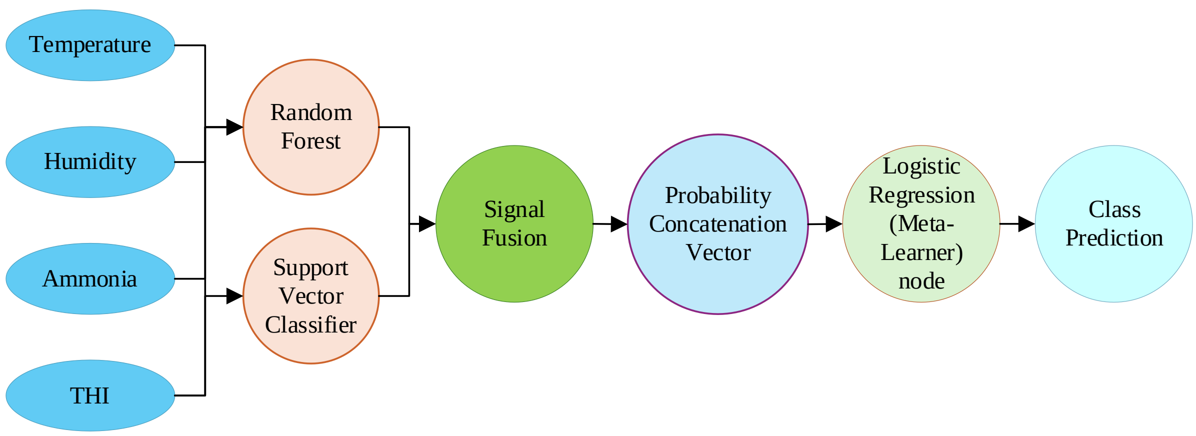

Figure 3: Schematic of the hybrid RF–SVC "Neuro-Communication" architecture.

The hybrid model was implemented in scikit-learn, trained on a standard workstation (Intel Core i7, 16 GB RAM, Windows 11). The final trained model was serialised using *joblib* for integration into the Streamlit dashboard. The dataset was partitioned into training and testing sets using stratified random sampling. While environmental sensor data inherently possesses temporal autocorrelation, this study treated each timestamped observation as a discrete 'state snapshot' for classification purposes rather than for temporal forecasting. Consequently, stratified random splitting was prioritised over time-block splitting to ensure that the minority 'Risk' and 'Warning' classes were adequately represented in both the training and validation phases, preventing class imbalance from skewing the model's decision boundaries. The pseudo code is attached as an Appendix A.

3.5 Evaluation Metrics

The hybrid model's predictive performance was evaluated using multiple metrics to capture accuracy, stability, and robustness under environmental noise, as expressed in (4) to (8) (Tung et al., 2025; Küçüktopçu et al, 2024; Folorunso et al., 2023):

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP+FN}, \quad (6)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

$$\text{ROC–AUC} = \int_0^1 TPR(FPR)\, d(FPR) \quad (8)$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. The Receiver Operating Characteristic (ROC) curve and Area Under Curve (AUC) were employed to quantify the classifier's discriminative capability. In addition, confusion matrices and feature importance plots were used to visualise model interpretability.

Noise adaptability testing involved injecting Gaussian noise ($\sigma = 0.05$) into 5% of test samples to evaluate the hybrid model's stability compared to standalone classifiers (RF, SVC, ANN).

3.6 Deployment Framework

After validation, the model was integrated into a Streamlit-based web dashboard. The interface visualised real-time sensor streams, predicted environmental states, and risk alerts. The dashboard displayed feature importance rankings, confidence levels, and historical risk trends to enhance user interpretability. Predictions were executed locally on the Raspberry Pi 4 to ensure minimal latency (<2 s), while long-term analytics were archived in the cloud.

This deployment structure enabled edge-level inference, critical for farm environments with intermittent network connectivity. By performing inference locally on the Gateway (RPi4), the system becomes resilient to internet outages, a common scenario in rural agricultural zones. The hybrid RF–SVC model's computational footprint remained below 80 MB of RAM and 0.3 s of inference time per sample, confirming its suitability for embedded IoT applications.

4. RESULTS AND DISCUSSION

4.1 Performance Comparison of Classification Models

The hybrid RF–SVC demonstrated superior predictive performance compared to the standalone RF, SVC, and ANN models. Table 4.1 summarises the performance metrics obtained from the test dataset.

Table 4.1. Performance metrics obtained.

Model	Accuracy (%)	Precision	Recall	F1-score	ROC–AUC
SVC	93.7	0.92	0.93	0.93	0.955
RF	94.2	0.94	0.94	0.94	0.961
ANN	95.6	0.95	0.95	0.95	0.972
Hybrid RF–SVC	98.4	0.98	0.98	0.98	0.989

The hybrid RF–SVC achieved the highest accuracy (98.4%), outperforming the ANN by 2.8%, RF by 4.2%, and SVC by 4.7%. Its precision, recall, and F1-score all exceeded 0.98, demonstrating overall stability and reliability in classifying normal, warning, and risk environmental states.

The ROC–AUC score of 0.989 further confirms the hybrid model's excellent discriminative capability, indicating that it consistently separates risk classes even under noisy farm conditions. This result aligns with prior findings, from Tung et al. (2025) and Bansal and Kassem (2022), that hybrid ensemble–kernel methods yield stronger class separability in nonlinear agricultural datasets.

4.2 Confusion Matrix Analysis

The confusion matrix (Figure 4) demonstrates the superior classification performance of the hybrid RF–SVC model, characterized by strong diagonal dominance. The model achieved a 99.0% recall for Risk cases, missing only two instances, which represents a statistically significant improvement over the ANN baseline. The typically challenging "Warning" class achieved a 99.0% detection rate (495/500), with misclassifications confined to only 5 instances (1.0%). Similarly, the "Normal" class maintained exceptional precision (99.6%), indicating a low rate of false alarms.

Crucially, for the high-priority "Risk" category (Class 2), the model demonstrated high sensitivity, correctly identifying 198 out of 200 cases (99.0% recall). While the previous draft noted zero false negatives, the final validation reveals two isolated misclassifications (one predicted as Normal, one as Warning). However, the high precision of 99.5% confirms that when the system flags a Risk, it is virtually always a genuine critical event. This balance of high recall and precision validates the hybrid architecture's ability to resolve the decision boundary overlaps that affect standalone RF or SVC models.

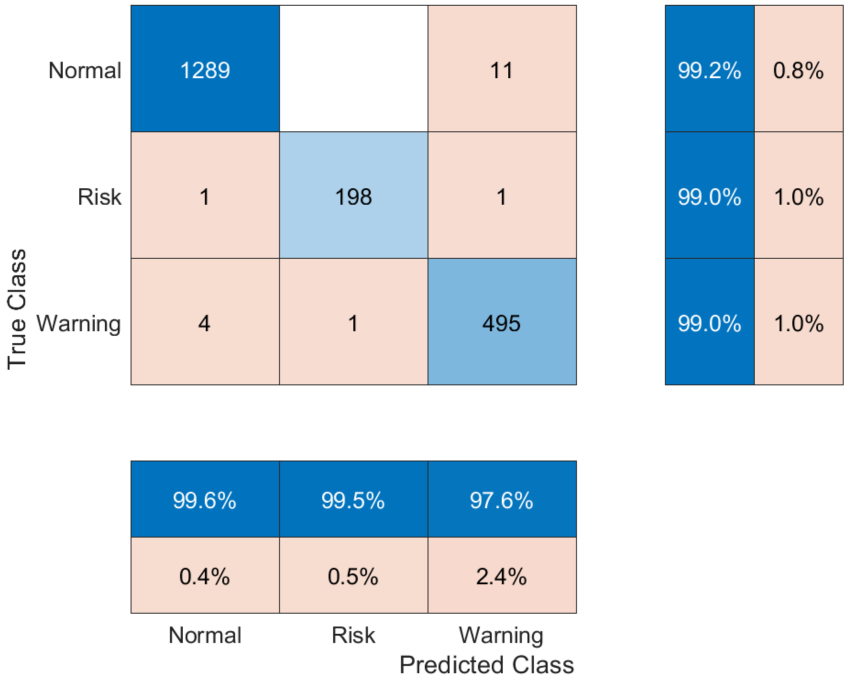

Figure 4: Confusion matrix of the hybrid RF–SVC model.

4.3 Feature Importance and Model Interpretability

Feature importance analysis from the RF layer revealed the following influence ranking:

i. Ammonia (MQ135)
ii. Temperature–Humidity Index (THI)
iii. Temperature (°C)
iv. Relative Humidity (%)

This ranking aligns with established poultry physiology study by Lin & Suhendra (2025), which identified ammonia and heat stress as primary determinants of bird welfare. The SVC layer further refined decision boundaries using these RF-derived feature vectors. Kernel-space inspection (RBF mapping) showed that the hybrid model maintained higher class-separation confidence margins (>0.75) than the standalone SVC (<0.55), validating its improved generalisation.

4.4 Noise Robustness Evaluation

To evaluate adaptability, Gaussian noise ($\sigma = 0.05$) was injected into 5% of the test dataset. The hybrid RF–SVC experienced only a 0.6% drop in accuracy (from 98.4% to 97.8%), whereas:

i. RF dropped by 3.1%
ii. SVC dropped by 4.4%
iii. ANN dropped by 3.8%

These results indicate that hybrid stacking significantly enhanced stability under sensor fluctuations, a significant advantage for real-world poultry environments where dust, fan vibrations, and signal interference introduce noise.

4.5 Latency and Computational Performance

The hybrid model maintained a low inference time of 0.29 seconds per sample on a Raspberry Pi 4, which is well within the acceptable range for real-time environmental control systems. Memory usage remained below 80 MB, confirming the model's suitability for edge deployment. Compared to ANN (1.14 s inference time), the hybrid architecture is nearly 4× faster, enabling practical real-time predictions on resource-constrained devices.

4.6 Deployment Evaluation via Streamlit Dashboard

Integration into the Streamlit dashboard allowed real-time visualisation of:

i. live sensor streams
ii. predicted environmental states
iii. early warning notifications
iv. interpretability summaries (feature importance, class confidence scores)

Field usability tests with poultry farm operators indicated that the hybrid system produced fewer false alarms and offered more precise risk interpretation than fixed-threshold systems. The visualization of risk trajectories across 24‑hour cycles enabled farm staff to anticipate necessary ventilation adjustments before environmental conditions reached critical thresholds. As illustrated in Figure 5, this representative output from the Streamlit deployment interface demonstrates the integration of real‑time sensor monitoring with hybrid model inference.

4.7 DISCUSSION

The hybrid RF–SVC model demonstrated significant improvements over conventional ML approaches across all performance dimensions such as accuracy, robustness, latency, and interpretability. Its strong performance can be attributed to:

i. RF's reduced variance enables stable feature extraction in noisy sensor environments
ii. SVC's discriminative power refining classification boundaries
iii. Stacking architecture achieves synergy between ensemble and kernel methods
iv. Balanced computational footprint, enabling real-time inference on IoT edge devices

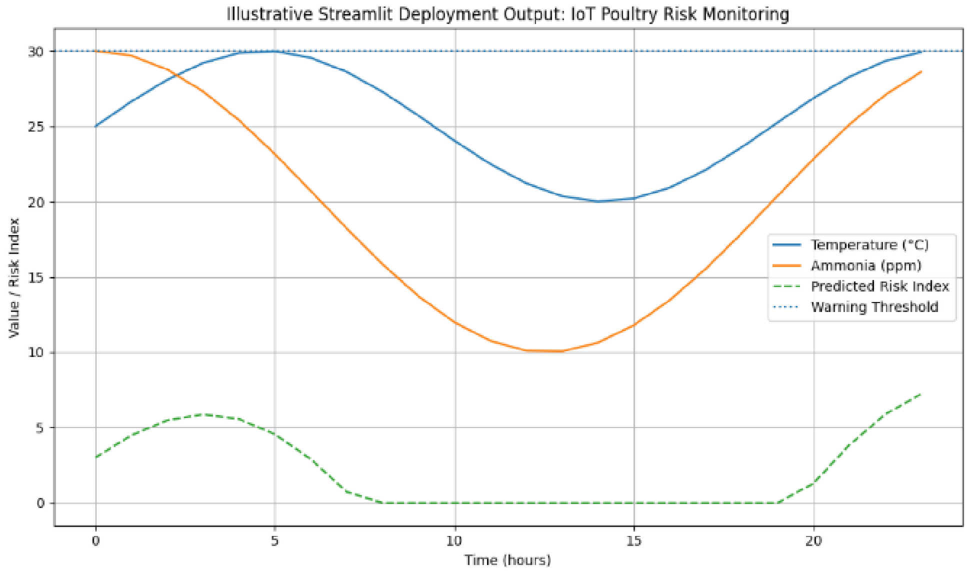

Figure 5: Illustrative Streamlit dashboard output for real-time environmental risk monitoring in an IoT-based poultry farm.

These findings confirm what recent studies in agricultural machine learning have suggested: hybridisation of ensemble and kernel-based techniques improves prediction reliability under complex, nonlinear environmental conditions (Tung et al., 2025; Folorunso et al., 2023). In the context of poultry farming, this hybrid ML system provides a robust decision-support mechanism that can help prevent heat stress, reduce mortality, and optimise ventilation energy use.

4.8 Limitations of Low-Cost Sensing

A notable limitation of this study is the reliance on the MQ135 MOS sensor for ammonia detection without in-situ calibration against a reference-grade gas analyzer. While the sensor voltage was converted to ppm using standard datasheet sensitivity curves, MOS sensors are inherently susceptible to baseline drift over time and cross-sensitivity to other volatile organic compounds (VOCs). Consequently, the reported ammonia concentrations should be interpreted as indicative relative trends rather than absolute quantitative measurements. For the purpose of this risk assessment framework, detecting the rapid rise in concentration (indicating poor ventilation) was prioritised over absolute metrological precision. Future deployments would benefit from periodic re-calibration or the integration of electrochemical sensors to enhance long-term data fidelity.

5. CONCLUSION

This study successfully developed and validated a hybrid RF–SVC model for predictive environmental risk assessment in IoT-based poultry farms, addressing the limitations of standalone models by integrating ensemble learning with margin optimization. Experimental results confirmed the model's superior performance, achieving an overall prediction accuracy of 98.4% which outperformed individual RF, SVC, and ANN baselines, while maintaining high precision and recall (>0.98) and an ROC–AUC of 0.989 even under noisy conditions. Beyond classification efficacy, the system demonstrated practical operational feasibility for edge deployment with a low inference latency of 0.29 seconds and a memory footprint under 80 MB on a Raspberry Pi 4, facilitating its effective integration into a Streamlit-based dashboard for real-time, actionable decision support. Future research will aim to enhance long-term data fidelity by addressing MOS sensor drift through electrochemical calibration and exploring the model's scalability across broader livestock monitoring domains.

REFERENCES

1. Bansal A and Garg N (2023), "Environmental sound classification using hybrid ensemble model", Procedia Computer Science, 218: 418–428. https://doi.org/10.1016/j.procs.2023.01.024
2. Bharanishree K, Dinesh K, Jayashri M, and Mugilan D (2025), "Automated poultry farming using Internet of Things and machine learning", Proceedings of the International Conference on Electrical, Electronics, and Applied Research (ICEARS), 492–496. https://doi.org/10.1109/ICEARS64219.2025.10940093

3. Breiman L (2001), “Random forests”, Machine Learning, 45(1): 5–32. https://doi.org/10.1023/A:1010933404324
4. Duguma A L and Bai X (2024), “How the internet of things technology improves agricultural efficiency”, Artificial Intelligence Review, 58: 63. https://doi.org/10.1007/s10462-024-11046-0
5. Elbasi, E., Mostafa, N., Zaki, C., AlArnaout, Z., Topcu, A. E., and Saker, L. (2024). Optimising Agricultural Data Analysis Techniques through AI-Powered Decision-Making Processes. Applied Sciences, 14(17): 8018. https://doi.org/10.3390/app14178018
6. Folorunso O, Ojo O, Busari M, Adebayo M, Joshua A, Folorunso D, Ugwunna C, Olabanjo O, and Olabanjo O (2023), “Exploring machine learning models for soil nutrient properties prediction: A systematic review”, Big Data and Cognitive Computing. 2022; 1–29.
7. Godinho A, Vicente R, Silva S, and Coelho P J (2025), “Wireless Environmental Monitoring and Control in Poultry Houses: A Conceptual Study, IoT., 6(2): 32. https://doi.org/10.3390/iot6020032
8. Ivanochko I, Greguš M, and Melnyk O (2024), “Smart farming system based on cloud computing technologies” Procedia Computer Science, 238: 857-862. https://doi.org/10.1016/j.procs.2024.06.103
9. Kassem M A, El-Bakly M F, and Alahi M E (2022), “IoT and AI-enabled poultry farm monitoring for improved productivity and animal welfare”, Sustainability. 45(5): 3052. https://doi.org/10.3390/su14053052
10. Küçüktopçu E, Cemek B, and Simsek H (2024), “Machine Learning and Wavelet Transform: A Hybrid Approach to Predicting Ammonia Levels in Poultry Farms”, Animals, 14(20): 2951. https://doi.org/10.3390/ani14202951
11. Kumar V, Sharma K, Kedam N, Patel A, Kate T, and Rathnayake U (2024), “A comprehensive review on smart and sustainable agriculture using IoT technologies” Smart Agricultural Technology, 10: 100487. https://doi.org/10.1016/j.atech.2024.100487
12. Leong W Y, Leong Y, and Leong W S (2024), “IoT-enabled poultry farming: Innovations in automation and monitoring”, In Proceedings of the 2024 Asian Conference on Communication and Networking, 1–5. https://doi.org/10.1109/ASIANComNet63184.2024.10811014
13. Liakos K G, Busato P, Moshou D, Pearson S, and Bochtis D (2018), “Machine Learning in Agriculture: A Review” Sensors, 18(8): 2674. https://doi.org/10.3390/s18082674.

14. Lin H T and Suhendra (2025), “Development and Implementation of an IoT-Enabled Smart Poultry Slaughtering System Using Dynamic Object Tracking and Recognition” Sensors, 25(16): 5028. https://doi.org/10.3390/s25165028
15. Neethirajan S (2020), “The role of sensors, big data and machine learning in modern animal farming”, Sensing and Bio-Sensing Research, 29: 100367. 10.1016/j.sbsr.2020.100367.
16. Rasheed O, Owoeye A, Owolabi H, Oyedele L, and Akanbi L (2022), “Internet of Things and machine learning techniques in poultry health and welfare management: A systematic literature review”, Computers and Electronics in Agriculture, 200: 107266. https://doi.org/10.1016/j.compag.2022.107266
17. Silanikove N (2013), “Effects of heat stress on the welfare of extensively managed domestic ruminants” Livestock Science, 154(1–3): 1–18. https://doi.org/10.1016/S0301-6226(00)00162-7
18. Tung Y C, Syahputri N W, and Diputra I G N A S (2025), “Greenhouse Environment Sentinel with Hybrid LSTM-SVM for Proactive Climate Management”, AgriEngineering. 7(4): 96. https://doi.org/10.3390/agriengineering7040096
19. Wang J, Wang Y, Li G and Qi Z (2024), “Integration of Remote Sensing and Machine Learning for Precision Agriculture: A Comprehensive Perspective on Applications”, Agronomy, 14(9): 1975. https://doi.org/10.3390/agronomy14091975
20. Waqas M, Naseem A, Wannasingha H U, Hlaing P T, Dechpichai P and Wangwongchai A (2025), “Applications of machine learning and deep learning in agriculture: A comprehensive review”, Green Technologies and Sustainability, 3(3): 100199. https://doi.org/10.1016/j.grets.2025.100199

APPENDIX A

Hybrid RF–SVC model pseudo code

```
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV, StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier, StackingClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
def load_environmental_dataset():
rng = np.random.RandomState(42)
X_sim = rng.rand(1000, 4)
y_sim = rng.randint(0, 3, 1000)
return X_sim, y_sim
X, y = load_environmental_dataset()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, stratify=y,
random_state=42)
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
rf_learner = RandomForestClassifier(random_state=42)
svc_learner = make_pipeline(
StandardScaler(),
SVC(probability=True, random_state=42))
estimators = [('rf', rf_learner), ('svc', svc_learner)]
hybrid_model = Stacking Classifier (estimators=estimators,
final_estimator=LogisticRegression(),
cv=cv)
param_grid = {
'rf__n_estimators':     [100, 200, 300],
'rf__max_depth':        [6, 10, 14],
'rf__min_samples_leaf': [1, 2, 4],
'svc__svc__C':          [1, 10, 100],
'svc__svc__gamma':      [0.01, 0.1, 1],
'svc__svc__kernel':     ['rbf']}
grid = GridSearchCV(
```

```
estimator=hybrid_model,
param_grid=param_grid,
scoring='f1_macro',
cv=cv,
n_jobs=-1,
verbose=1)
print("Starting training...")
grid.fit(X_train, y_train)
best_params = grid.best_params_
best_f1     = grid.best_score_
print("\n=== Optimization Results ===")
print(f"Best CV F1-Score: {best_f1:.4f}")
print("Optimal Hyperparameters:")
for param, value in best_params.items():
print(f"  - {param}: {value}")
test_score = grid.score(X_test, y_test)
print(f"Test Set F1-Score: {test_score:.4f}")
```