

Original software publication

AgriAlertX: Climate-driven disaster prevention for agriculture

Dadi Soukaina ^{a,b,*}, Lachgar Mohamed ^{b,c}, Fattouhi Radwa ^d, Doudy Sifeddine ^d,
Hamid Hrimech ^e, Kartit Ali ^a

^a Chouaib Doukkali University, National School of Applied Sciences, LTI Laboratory, El Jadida, Morocco

^b Cadi Ayyad University, Higher Normal School, Computer Science Department, Marrakech, Morocco

^c Cadi Ayyad University, L2IS Laboratory, Marrakech, Morocco

^d Chouaib Doukkali University, National School of Applied Sciences, TRI Department, 2ITE, El Jadida, Morocco

^e Hassan First University, National School of Applied Sciences, LAMSAD Laboratory, Berrechid, Morocco

ARTICLE INFO

Keywords:

Agricultural monitoring
Disaster prevention
Analysis of meteorological data
Agricultural alerts

ABSTRACT

Climate change has led to increasingly unpredictable weather events, posing serious threats to agricultural productivity worldwide. AgriAlertX is a digital decision-support platform designed to strengthen climate resilience for farmers through real-time alerts and context-specific recommendations. By integrating open weather APIs, predictive analytics, and crop-specific thresholds, the system delivers localized guidance tailored to each user's geolocation and crop type. The platform helps mitigate the effects of droughts, frost, and heatwaves by supporting proactive decision-making and efficient resource use. Preliminary evaluations suggest that AgriAlertX may reduce agricultural losses by up to 20%, depending on crop sensitivity and regional vulnerability. This initiative contributes to improved risk management and supports global efforts in climate-resilient, sustainable agriculture.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-25-00092
Permanent link to Reproducible Capsule	N/A
Legal Code License	MIT License
Code versioning system used	Git
Software code languages, tools, and services used	Python, Flask, Kotlin, Java, Spring Boot, Swift, SwiftUI, TypeScript, Next.js
Compilation requirements, operating environments & dependencies	Android Studio, Python 3.x (transformers, flask), XCode, MySQL, Docker
Link to developer documentation/manual	https://github.com/Radwa-f/AgriAlertX/blob/master/README.md
Support email for questions	m.lachgar@uca.ac.ma

1. Motivation and significance

Agriculture faces growing vulnerability due to climate variability and extreme events — droughts, floods, heatwaves, and pest outbreaks — that cause substantial economic losses estimated at over \$12 billion annually [1]. Smallholder farmers, producing about one-third of the world's food, are disproportionately impacted by these disruptions because they have limited access to reliable weather forecasts and adaptive technologies [2]. For example, in arid and semi-arid regions, prolonged droughts have led to 20%–40% declines in cereal yields over the past decade [3], exacerbating food insecurity.

AgriAlertX responds to these challenges by delivering early warnings and actionable, crop-specific recommendations based on localized weather data. Unlike conventional tools that focus on single variables such as soil moisture or irrigation, AgriAlertX integrates real-time meteorological information with adaptive decision models tailored to crop type and location. This enables the platform to provide optimized irrigation advice and planting schedules in drought-prone areas, while also triggering flood or frost protection protocols when needed, enhancing both anticipatory capacity and adaptive response [4].

* Corresponding author.

E-mail address: soukaina.d623@ucd.ac.ma (D. Soukaina).

Reactive strategies alone are often insufficient for preventing major yield losses. AgriAlertX shifts to a proactive approach, issuing alerts based on predictive indicators so that farmers can adjust practices before critical thresholds are breached. Advance warnings, for example, enable timely adoption of water-saving or soil improvement techniques, reducing potential losses by up to 25% [3].

As global population nears 9.7 billion by 2050 [5], protecting agricultural productivity under climate stress becomes imperative. AgriAlertX supports this through precision forecasting and localized interventions that help farmers maintain yields and use inputs efficiently, aligning with Sustainable Development Goal 2 (Zero Hunger).

To address the limitations of existing platforms, AgriAlertX combines adaptive analytics, participatory validation, and broad accessibility. Its core dynamically calibrates crop stress thresholds with real-time weather and phenological data, refines alerts using feedback from farmers and agronomists, and delivers support through a multilingual, cross-platform interface with integrated AI-powered chatbot. The platform incorporates authoritative crop-climate datasets from sources such as the FAO, USDA, and others, and maintains robustness with continuous integration and static code analysis. Its holistic feature set — including weather alerts, pest and disease warnings, predictive analytics, and agricultural news — positions AgriAlertX as a comprehensive solution for smallholder farmers in climate-vulnerable regions (Table 2).

2. Software description

This project delivers a user-friendly platform focused on real-time decision-making and proactive risk management in agriculture. By analyzing location-specific weather data and integrating crop-specific recommendations, the system enables farmers to make informed decisions, protect yields, and minimize losses. Advanced analytics and predictive modeling provide real-time updates and actionable insights, helping users adapt to adverse conditions, optimize resource allocation, and build resilience against climate change. Ultimately, AgriAlertX supports sustainable farming and food security.

2.1. Software architecture

The AgriAlertX platform is built with a modern, flexible, and strong software design that meets the needs of agricultural resilience and fast decision-making. This section delineates the architectural selections and their justifications, illustrating their contributions to system efficacy and user satisfaction (see Fig. 1).

1. Overview of the Architecture

AgriAlertX employs a microservices-based architecture to enhance scalability, modularity, and maintainability. Each component, from mobile and web clients to the chatbot, operates independently but communicates seamlessly with the backend through RESTful APIs. The architecture is containerized using Docker [6], ensuring consistent development, testing, and production environments. This modular design allows for efficient updates and minimal downtime, making the platform flexible and reliable for diverse user needs.

2. Backend Infrastructure

The backend infrastructure is designed to be robust, secure, and scalable to handle real-time agricultural data processing. Several key technologies and frameworks are utilized to achieve this goal:

- **Spring Boot Framework:** The backend is built with Spring Boot [7] due to its simplicity in setting up production-ready applications. Its robust dependency injection, support for RESTful APIs, and modular design make it an ideal choice for scalable back-end development. It is the heart connecting all components of the system.

- **Spring Security:** User authentication and data protection are handled by Spring Security [8], which provides advanced mechanisms for secure authorization and transaction handling.
- **Persistence Layer:** The backend leverages JPA and Hibernate [9] for object-relational mapping, simplifying database operations while ensuring high performance. MySQL [10] is used for data storage, ensuring scalability and reliability.
- **Flask for Chatbot Services:** Flask [11] powers the chatbot microservice, providing lightweight REST APIs and efficient communication with the Hugging Face AgriQBot model [12].
- **Python for Crops Analysis Services:** Scikit-Learn powers the python model, and FastApi provides lightweight REST API and efficient communication for an optimal crops analysis and recommendation [13].

3. Frontend Composition

The frontend is designed to ensure a seamless user experience across web and mobile platforms. It incorporates modern frameworks and best practices to achieve a responsive and interactive interface.

- **Next.js for Web Client:** The web interface uses Next.js [14] for server-side rendering and dynamic routing, ensuring a fast, SEO-optimized user experience. Its component-based architecture simplifies development and enhances maintainability.
- **Kotlin for Android:** The Android client employs Kotlin [15], which offers concise syntax, null safety, and seamless integration with modern Android frameworks. It follows an MVVM architecture, with LiveData for reactive programming and Retrofit [16] for API communication.
- **Swift for iOS:** The iOS client is developed using Swift [17], offering optimized performance and seamless integration with Apple's ecosystem. The application integrates Alamofire [18] for efficient HTTP networking.
- **Responsive Design:** All interfaces are designed to be responsive, ensuring accessibility across various devices, including smartphones, tablets, and desktops.

4. Database Management

Effective database management is crucial for ensuring data integrity, security, and performance. The AgriAlertX platform employs the following technologies to optimize data storage and retrieval:

- **MySQL:** The choice of MySQL [10] for database management is driven by its reliability, performance, and ability to handle complex queries. Its scalability and cost-effectiveness make it ideal for agricultural data storage.
- **Hibernate ORM:** Hibernate [9] simplifies database interactions by mapping the object-oriented domain model to the relational database, reducing boilerplate code and improving data management efficiency.

5. Communication and Data Flow

The platform employs RESTful APIs to facilitate seamless communication between the backend and frontend components. Data is exchanged in lightweight JSON format for compatibility across platforms. Each client (Android, iOS, and web) handles HTTP requests using specialized libraries like Retrofit (Android) [16], Alamofire (iOS) [18], and Axios [19] and Fetch [20] (web). This consistent communication architecture ensures real-time data flow and an optimal user experience.

6. Deployment and Quality Assurance

AgriAlertX is deployed as a containerized stack: the Android app communicates with a Spring Boot API gateway, which fetches weather data, normalizes inputs, and interacts with a Python microservice for hybrid analysis. The engine combines crop-specific rule checks with a multinomial logistic-regression model trained on the Kaggle crop-recommendation dataset (using temperature, humidity, rainfall, and pH as features, with fallback when pH is missing).

Docker [6] images are used consistently across development, staging, and production, with CI/CD pipelines for automated building, testing, and deployment. System health is monitored via dedicated endpoints, logs, metrics, and SLO alarms, with secrets managed securely and only coarse location data stored. Quality assurance leverages SonarQube [21] to ensure code quality, security, and maintainability. Comprehensive unit, integration, and end-to-end tests are run for each component to uphold high performance and reliability standards.

7. Innovative Components

To enhance user experience and decision-making capabilities, the AgriAlertX platform integrates several innovative features:

- **Chatbot Integration:** Implemented using a Flask API and the AgriQBot model from Hugging Face [12], the chatbot leverages a sequence-to-sequence (Seq2Seq) architecture to generate responses based on user queries. It uses an encoder-decoder framework, where the encoder processes input into a context vector capturing semantic meaning and the decoder generates step-by-step responses. An attention mechanism further enhances accuracy by focusing on specific input parts during response generation. While the chatbot currently provides basic agricultural assistance, its capabilities can be expanded through fine-tuning on domain-specific datasets to handle more complex queries effectively.
- **Predictive Analytics and Threshold Analysis:** The platform integrates a weather prediction API provided by Open-Meteo [22] and a crop-specific threshold analysis to deliver actionable insights, enabling farmers to take proactive measures against adverse conditions.
- **Real-Time Notifications:** Instant alerts are delivered to users via mobile and web platforms, ensuring timely actions to mitigate potential risks.
- **Agricultural News:** As an additional feature, AgriAlertX provides farmers the possibility of receiving weekly news related to agriculture by implementing the World News API [23].
- **State Management in Web Client:** The web client employs React hooks like `useState` and `useEffect` for efficient state management, resulting in a responsive and user-friendly interface.

2.2. Software functionalities

The AgriAlertX platform offers a comprehensive set of user-focused features to enhance agricultural resilience and manage disaster risks. Its functionalities are designed to optimize user experience, enable timely decisions, and support effective crop management:

1. **User and Location Management:** Secure user accounts allow farmers to register, specify key crops, and receive personalized notifications and recommendations. Users can update personal details, manage crop lists, and set or adjust their geographic location, either via geolocation or manual entry, ensuring tailored, localized advice for each farm.

2. **Weather Monitoring:** The platform delivers real-time weather data — temperature, precipitation, and forecasts — so farmers can adapt activities accordingly. For instance, a forecast of heavy rain may prompt postponing irrigation to avoid overwatering. Access to historical weather trends helps refine long-term strategies.
3. **Alert System:** Automated alerts notify users about hazards such as frost or drought, based on local meteorological data and crop type. Prompt notifications allow timely protective measures (e.g., covering crops before frost), and farmers can access alert history for review and planning. The BPMN diagram (Fig. 3) illustrates these core routines.
4. **Agricultural Recommendations:** The system provides tailored advice on irrigation, fertilization, and preventive measures, dynamically adjusted for each crop and weather scenario. For example, during drought, it may suggest modified watering schedules for maize to conserve water while maintaining yield.
5. **Agricultural News:** An integrated news module keeps farmers updated on industry trends and best practices, with the ability to browse, search, and share relevant articles, supporting ongoing learning and community collaboration.
6. **Chatbot Assistance:** An interactive AI-powered chatbot answers farmer questions in real time using natural language processing. For example, it may recommend optimal wheat irrigation times in winter or suggest organic pest control solutions during rainy periods, providing immediate, actionable guidance.

Together, these features, including user and location management, real-time weather, alerts, recommendations, news, and chatbot support, work together to deliver actionable insights, enabling farmers to proactively manage crops and respond effectively to changing conditions.

2.3. Analysis and recommendation engine

The AgriAlertX decision engine combines deterministic agronomy with machine learning to provide both transparent explanations and predictive insights. At runtime, the mobile app sends user crops and location to the backend, which resolves forecasts and forwards relevant features (max/min temperature, rainfall extremes, mean humidity) to a Python analysis endpoint.

The engine consists of two layers:

1. **Rule-based layer:** Using crop tolerance profiles (min/max temperature, rainfall, humidity), it calculates deviations from optimal ranges and generates severity scores (LOW/MEDIUM/HIGH). Targeted recommendations are produced from an auditable knowledge base, ensuring explainability for each suggested action. Crop thresholds are sourced from FAO Crop Information [24], Ecocrop [25], and USDA Plant Hardiness [26].
2. **Machine learning layer:** A lightweight Python microservice runs a scikit-learn pipeline (standardization + multinomial logistic regression) trained on the Crop Recommendation dataset (features: temperature, humidity, rainfall, pH). When the model's confidence exceeds 90% and matches a user's crop, the UI highlights ideal conditions and proactive tips; otherwise, the system defaults to rule-based results. Model health is checked continuously; if unavailable or below quality thresholds, the fallback is automatic and seamless.

The orchestration, shown in the BPMN diagram (Fig. 3), routes requests through “ML ready?” and merges outputs into a uniform response for the app. This modular design allows updates to weather providers, tolerance tables, or recommender templates independently.

Model evaluation (Table 1, Fig. 2) shows robust performance: hold-out accuracy 0.79 (true pH) and 0.73 (assumed pH), macro-F1 of 0.79 and 0.70 respectively, with top-3 accuracy ≈ 0.95 . The confusion matrix illustrates crop class separation and informs threshold settings.

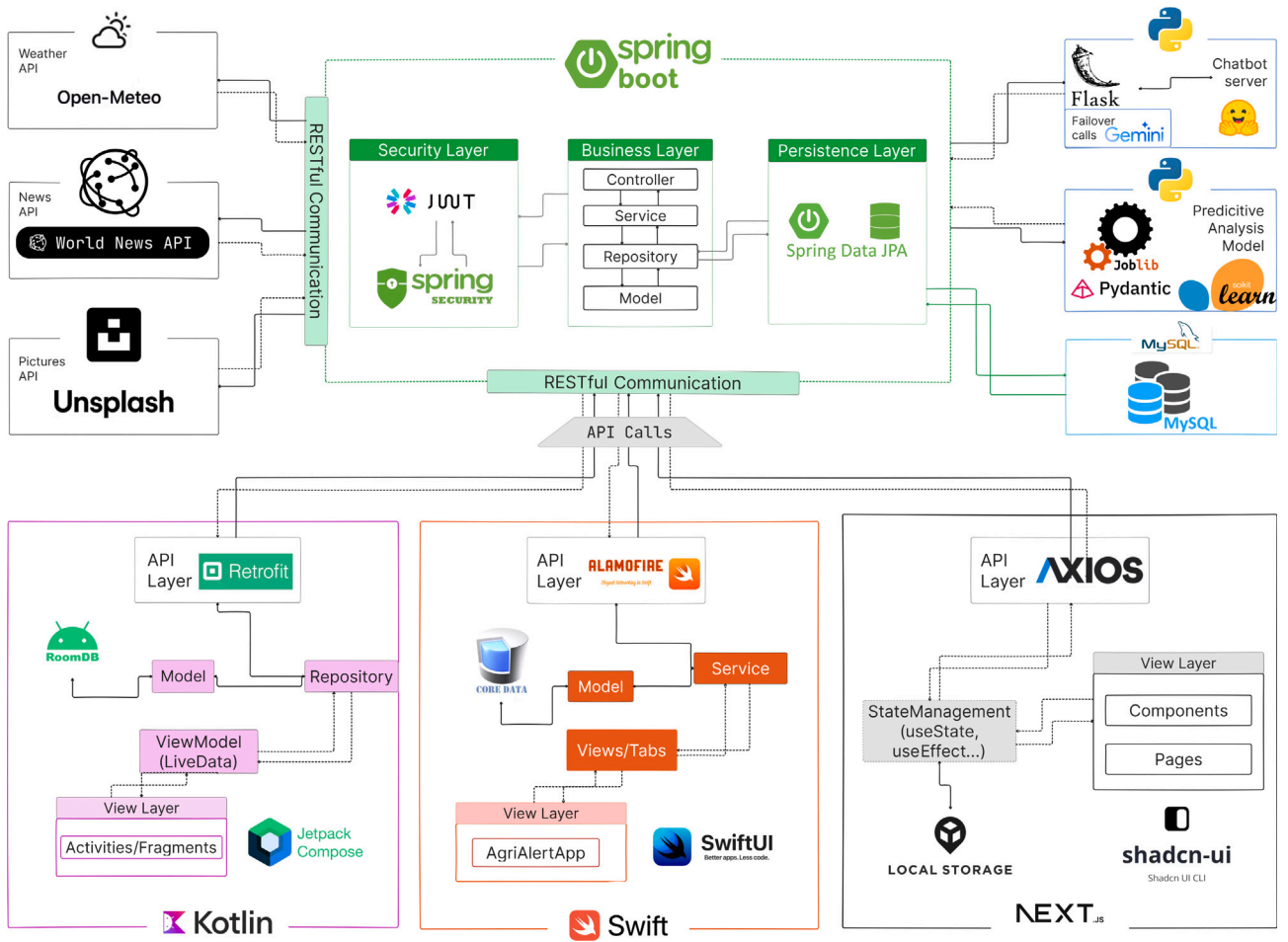


Fig. 1. AgriAlertX architecture.

Table 1

System performance metrics.

Setting	Accuracy	Macro-F1	Top-3 Accuracy
Hold-out (true pH)	0.793	0.787	0.957
Hold-out (assumed pH = 6.5)	0.732	0.696	0.950
5-fold CV (true pH)	0.782 ± 0.010	0.774 ± 0.012	–

3. Illustrative examples

To better understand how *AgriAlertX* assists farmers in mitigating climate-related risks, we give many illustrative scenarios that showcase its practical applicability. These instances illustrate how the platform delivers real-time notifications, tailored suggestions, and practical insights to facilitate agricultural decision-making and guarantee prompt solutions. Moreover, *AgriAlertX* integrates many supplementary features to enhance use and accessibility, hence advancing agricultural risk management.

3.1. Extreme weather alerts for crops

Agricultural productivity is highly vulnerable to extreme events such as drought, heat stress, frost, and cold stress. These reduce yields, heighten pest risks, and threaten food security. To address this, the platform employs an ML-powered early warning system that combines real-time weather data, predictive analytics, and crop-specific thresholds from agronomic literature (e.g., [27]). Phenology-aware indices such as Growing Degree Days (GDD) and water balance indicators refine alert timing, while mobile alerts with severity markers and

recommendations guide farmers in real time. Beyond adverse events, the system also identifies favorable growing conditions when forecasts align with crop tolerance profiles at high confidence (see Figs. 6(a), 6(b)).

The following scenarios highlight how the system delivers timely, crop-specific guidance. While this section focuses on coconut, wheat, and grapevine, future validation will extend to other climate-sensitive crops such as rice and maize.

3.1.1. Scenario 1: Coconut drought and heat stress

Coconut production is highly sensitive to drought and elevated temperatures, which reduce yields, deplete water resources, and increase pest risks. When critical thresholds for temperature and rainfall are exceeded, the system issues high-severity alerts and delivers localized drought insights (see Figs. 4(a), 4(b)).

To mitigate impacts, farmers are guided to optimize irrigation, apply mulching, and adjust soil fertility management (see Fig. 4(c)). These measures enhance resilience and help safeguard yields under prolonged stress (see Fig. 4).

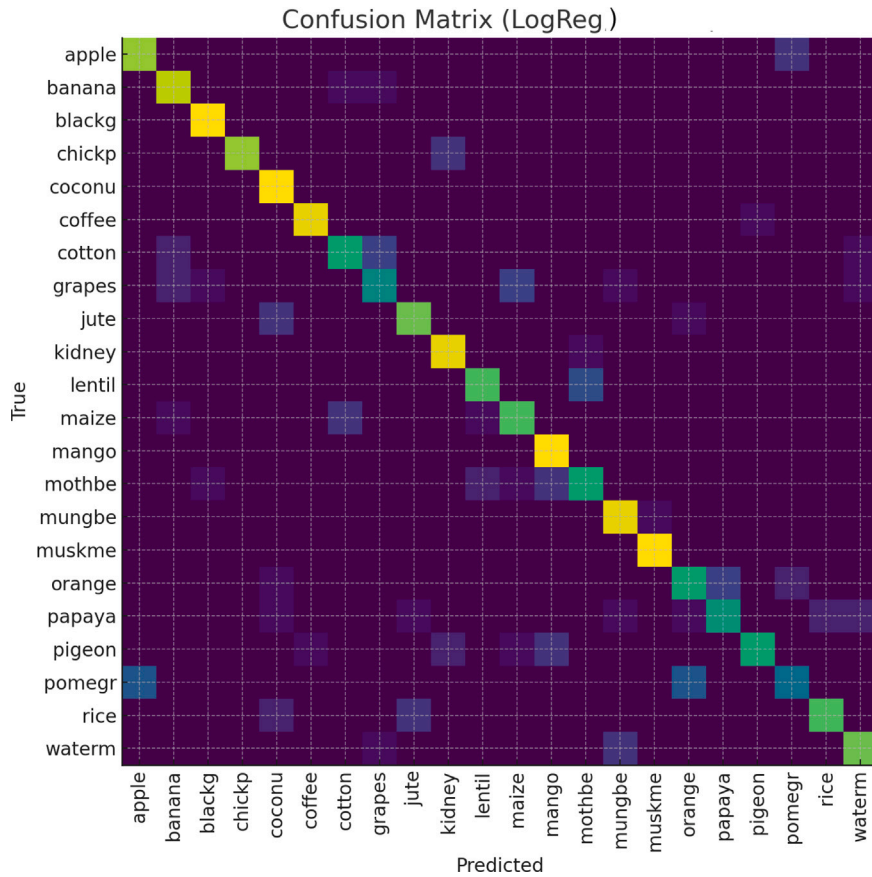


Fig. 2. Crops confusion matrix.

3.1.2. Scenario 2: Wheat frost and cold stress

Wheat crops are vulnerable to frost and cold stress, which damage tissues, reduce germination, and slow growth, particularly under low soil moisture. The system issues medium-severity alerts when minimum temperatures fall below thresholds adjusted for phenological stage [27]. Insight notifications highlight temperature anomalies and frost risks (see Figs. 4(d), 4(e)).

Recommended practices include nighttime irrigation to buffer soil temperature, organic mulching to limit heat loss, and moisture conservation to reduce cold-induced drought stress (see Fig. 4(f)). These targeted measures help maintain productivity under extreme cold (see Fig. 4).

3.1.3. Scenario 3: Grapevine heatwave and low rainfall

Grapevine growth is threatened by heatwaves combined with low rainfall, which accelerate sugar accumulation, cause berry shriveling and sunburn, and limit nutrient uptake. The system issues high-severity alerts when temperature and rainfall thresholds are exceeded, supported by GDD and rainfall deviation indices. Insight panels highlight dual-stress risks and flag critical stages such as flowering and ripening (see Figs. 5(a), 5(b)).

Mitigation strategies include regulated deficit irrigation or supplemental watering, mulching and cover crops to conserve soil moisture, and canopy management practices such as strategic leaf removal (see Fig. 5(c)). These recommendations help preserve both yield and fruit quality under extreme conditions (see Fig. 5).

3.2. Additional features

In addition to real-time weather alerts, *AgriAlertX* offers advanced features to improve decision-making, boost efficiency, and support farmers in daily operations. These tools provide real-time insights,

personalized assistance, and geolocated services for more informed and responsive practices.

A key component is the AI-powered chatbot, which delivers context-specific advice on crop protection, irrigation, and soil health (see Fig. 6(d)). Its recommendations adapt to crop type and weather conditions for precision and relevance.

To increase alert accuracy, the platform integrates geolocation, allowing users to set locations manually or via GPS. This ensures highly localized alerts — such as climate forecasts, frost warnings, and drought advisories — tailored to each area. Multiple locations can be saved for monitoring distributed plots (see Fig. 6(e)).

The system also provides real-time notifications on weather risks, crop health, and personalized action plans, helping farmers adjust irrigation and respond quickly to changing conditions (see Fig. 6(f)).

By combining smart insights, location-aware analytics, and instant communication, the platform delivers a comprehensive decision-support system that enhances precision and climate resilience across agriculture.

4. Impact

The integration of real-time monitoring, AI-based analytics, and crop-specific recommendations in *AgriAlertX* strengthens both agricultural resilience and decision-making. By combining machine learning, satellite data, and IoT sensors, the platform improves weather forecasting, soil moisture tracking, and crop health assessment, supporting precision farming, disaster preparedness, and sustainable agriculture [28, 29].

In practice, *AgriAlertX* empowers farmers to anticipate and respond to climate threats through real-time alerts and tailored advice, enabling efficient input use and reducing both costs and potential

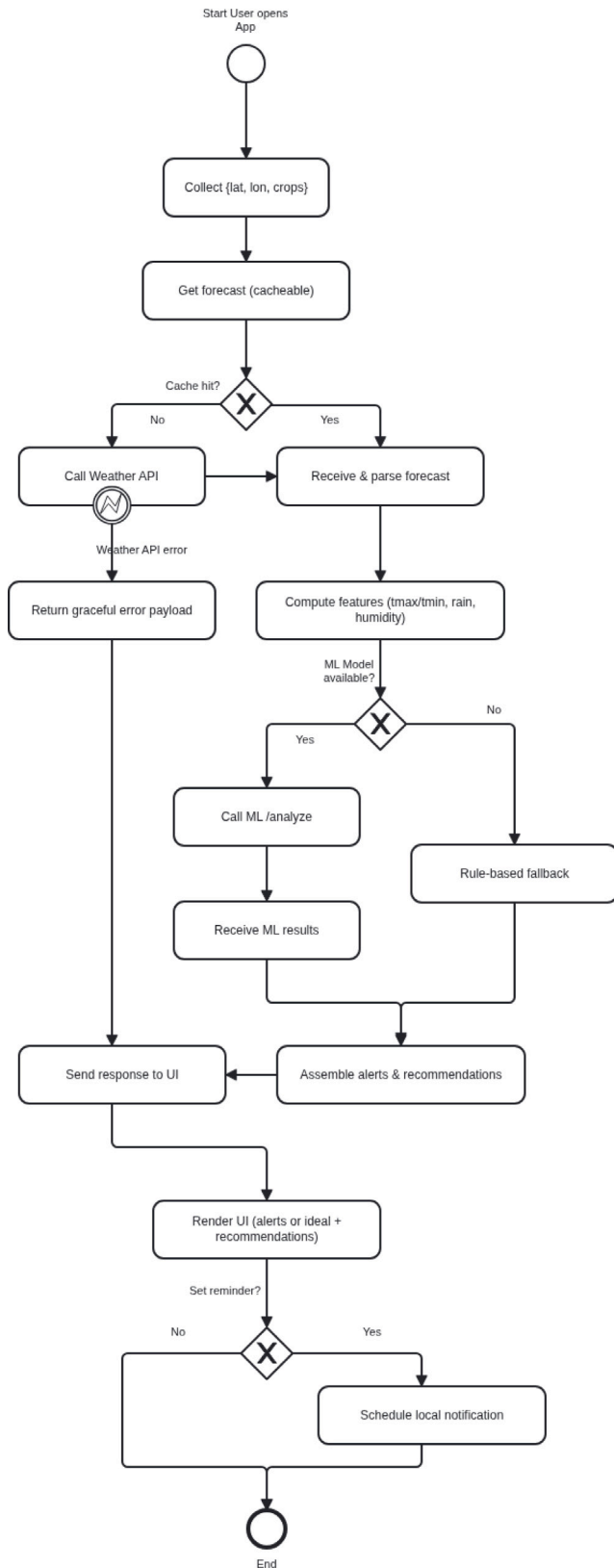


Fig. 3. AgriAlertX BPMN workflow diagram.

losses [30,31]. Its integrated AI-powered chatbot delivers expert guidance on pest management, planting schedules, and soil health, making scientific insights accessible even to smallholder farmers, and ensuring recommendations are relevant and timely through location-aware monitoring.

From a policy perspective, AgriAlertX supports food security and climate adaptation by promoting sustainable land management. It offers actionable insights on climate variability, water use, and soil conservation, helping farmers reduce chemical overuse, limit water waste, and adopt conservation practices [32]. The platform's datasets also aid policymakers and agencies in developing evidence-based adaptation frameworks and rural development strategies [33].

Overall, AgriAlertX delivers impact from farm-level resilience to scientific and policy innovation, empowering users with actionable intelligence to support stable, productive, and sustainable food systems.

4.1. Comparison with existing tools

Several existing tools support climate resilience and disaster mitigation by offering weather insights and agronomic recommendations. However, the present solution differentiates itself through its unique combination of real-time predictive alerts, AI-powered diagnostics, and actionable guidance.

As detailed in Table 2, tools like CropX and Climate FieldView focus on optimizing irrigation and crop health using weather-based data. While CropX integrates soil monitoring and tailored recommendations, Climate FieldView emphasizes AI-powered yield forecasts. Yet both systems primarily emphasize monitoring over mitigation.

Other tools—such as Granular and FarmLogs offer general farm management functions with limited crop-specific analytics and lack proactive pest or disease alerts. Although Taranis employs AI for yield prediction, it omits critical features like real-time weather warnings and soil health assessments.

The present solution unifies weather forecasting, AI analytics, and immediate notifications to support operational readiness. Unlike AgriSync, which connects farmers to human experts, this tool offers autonomous, automated guidance. Compared to FarmLogs, which delivers generalized forecasts, it improves on-the-ground responsiveness through precise and localized decision support.

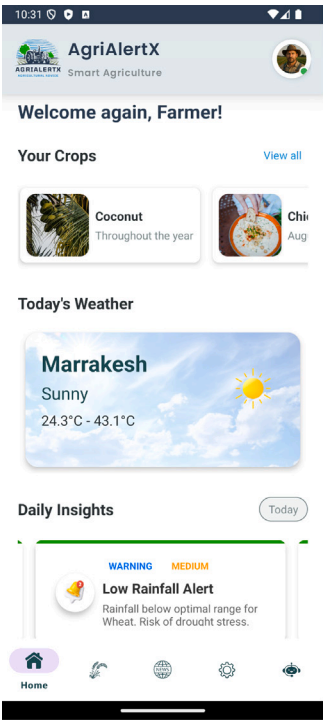
Currently in its prototype phase, the system is accessible via iOS, Android, and web platforms. With its integrated features—such as pest and disease alerts, predictive yield modeling, and IoT connectivity it holds strong potential to reshape agricultural adaptation and climate resilience in both industrial and smallholder farming contexts.

5. Collaborative validation and future directions

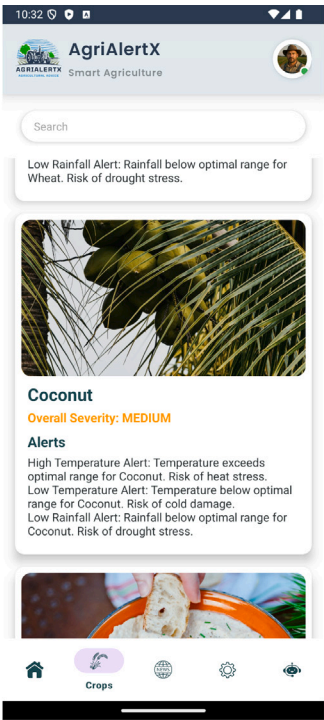
To ensure the platform's robustness, scalability, and applicability, AgriAlertX will undergo collaborative validation. While the prototype shows strong potential for agricultural resilience, extended testing across diverse agro-ecological zones and crop types remains essential to refine predictive models and thresholds.

A key direction is a crowdsourcing framework where farmers, agronomists, and institutions contribute field observations and contextual feedback. This participatory loop enables real-time calibration of agroclimatic thresholds such as temperature and precipitation extremes based on local crop responses [34]. Such collaboration not only strengthens scientific validity but also accelerates the evolution of mitigation strategies and system adaptability. A dedicated app interface will allow users to annotate alerts, confirm or reject recommendations, and submit feedback on crop performance, fostering transparency and community ownership.

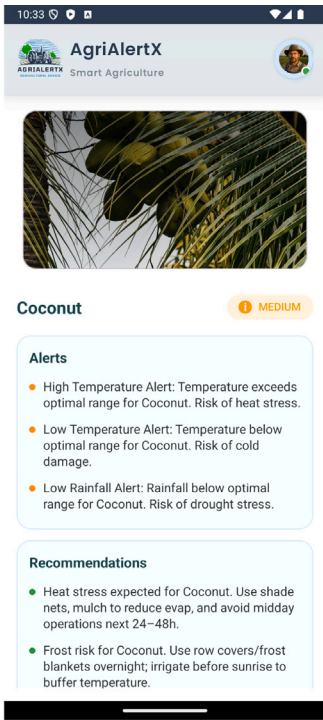
Future developments will integrate open agricultural datasets, extension systems, and research networks, aligning AgriAlertX with global efforts in open, participatory digital agriculture. This vision positions



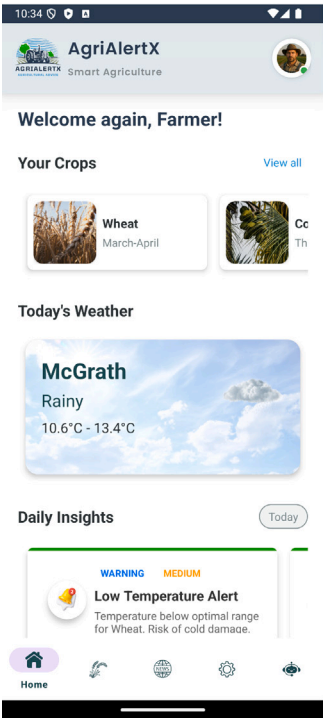
(a) Insight Alert



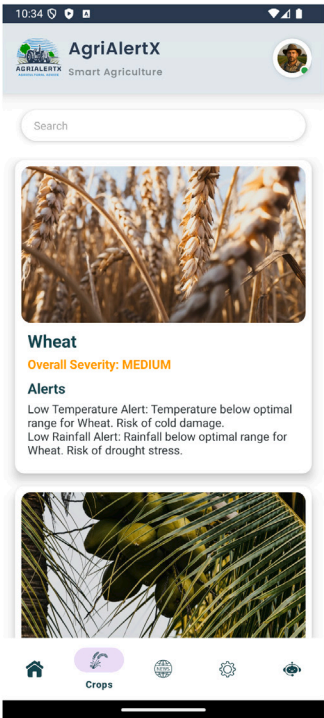
(b) Crops Alert



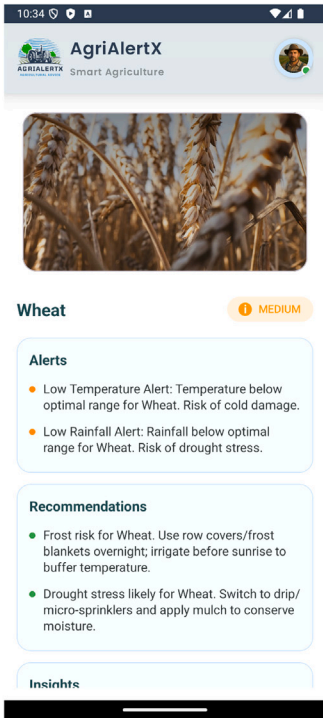
(c) Drought Recommendations



(d) Wheat – Insight Alert



(e) Wheat – Crops Alert



(f) Wheat – Recommendations

Fig. 4. AgriAlertX android application: Drought and heat stress alert for coconut and wheat.

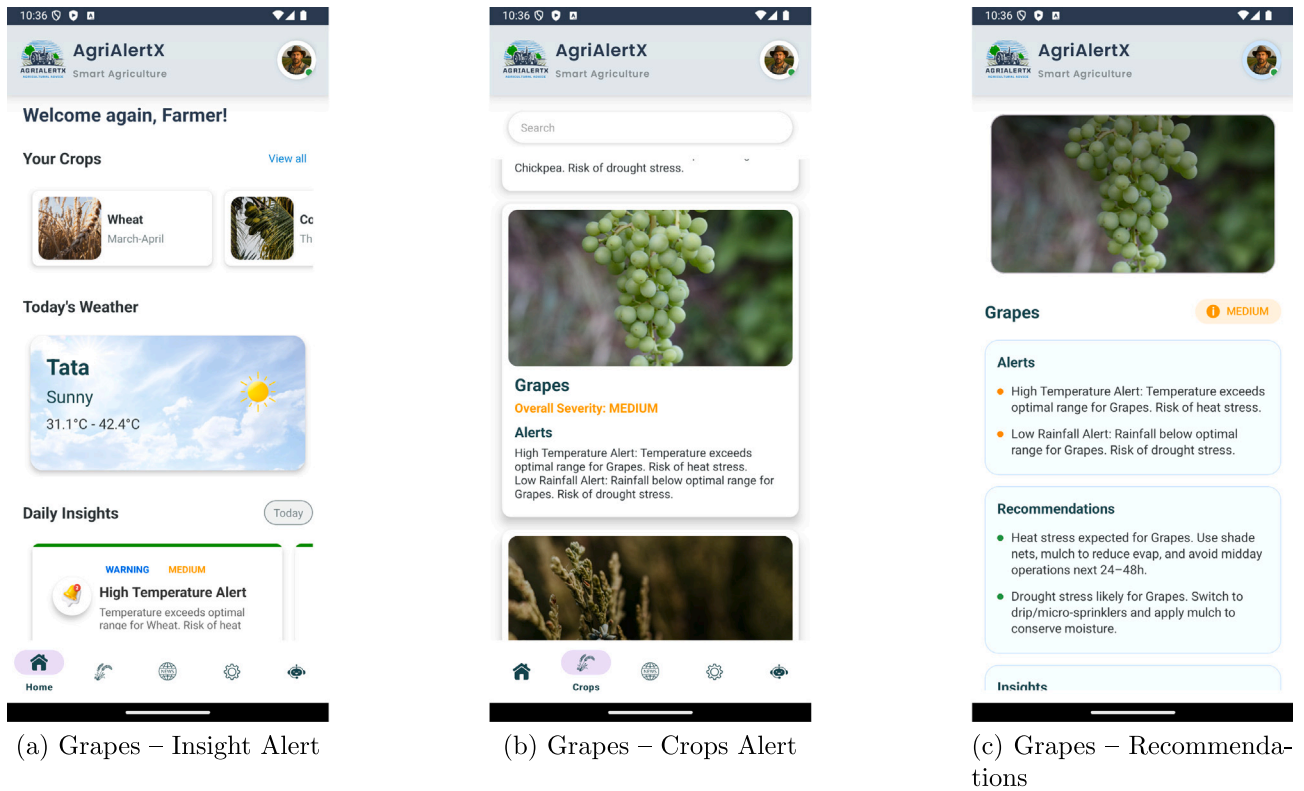


Fig. 5. AgriAlertX android application: Cold stress alerts with recommendations for Grapes.

Table 2

Comparative analysis of agricultural tools.

Feature	AgriAlertX	CropX ^a	Climate FieldView ^b	Granular ^c	Taranis ^d	FarmLogs ^e
Real-Time Weather Alerts	Yes	Limited	Yes	No	No	Limited
Crop-Specific Recommendations	Yes	Yes	No	Limited	Yes	Limited
AI-Driven Analysis	Yes	Yes	Yes	No	Yes	No
Soil Health Monitoring	Limited	Yes	No	Yes	No	No
Predictive Yield Forecasting	Yes	Limited	Yes	Yes	Yes	Yes
Pest and Disease Alerts	Yes	No	No	No	Yes	No
User-Friendly Mobile Interface	Yes	Yes	Yes	Limited	Yes	Yes
Integration with IoT Devices	Moderate	Yes	Yes	Yes	Yes	Limited
Cost Efficiency	High	Moderate	Moderate	Moderate	Low	High
Focus on Smallholder Farmers	High	Low	Moderate	Low	Moderate	High

^a <https://www.cropx.com>.

^b <https://www.climate.com>.

^c <https://www.granular.ag>.

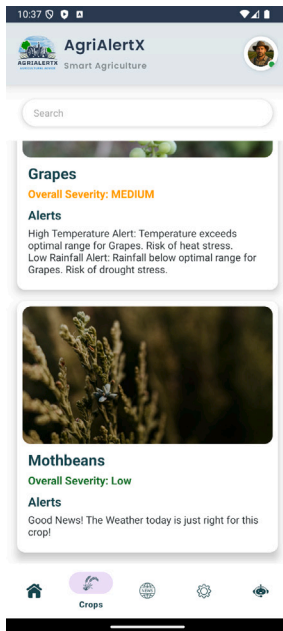
^d <https://www.taranis.ag>.

^e <https://www.farmlogs.com>.

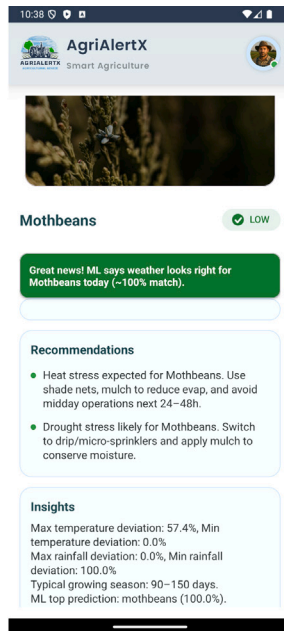
the platform as a scalable and sustainable tool for climate resilience and food security.

To broaden generalizability, future iterations will incorporate models for grapevine, rice, and maize selected for scientific and regional relevance thus enhancing the diversity of stress profiles and supporting

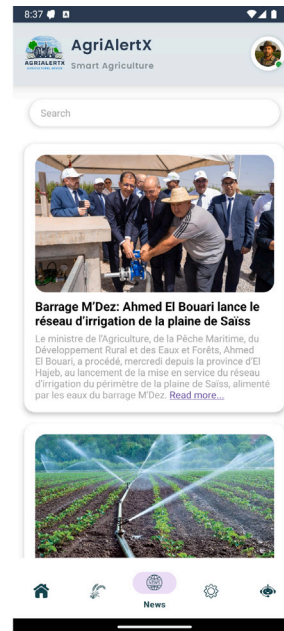
wider deployment. The participatory validation will be operationalized via a mobile crowdsourcing module, enabling users to upload standardized field observations (timestamp, GPS, crop type, growth stage, stress symptoms, local weather, mitigation actions) through a secure backend API.



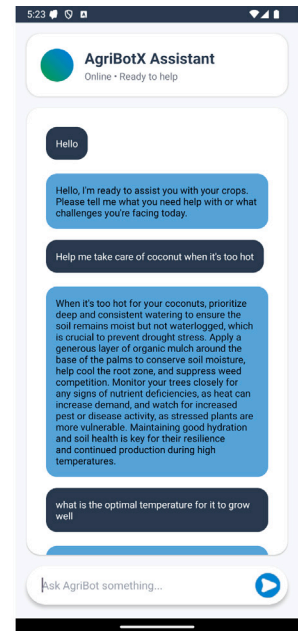
(a) Right weather



(b) Mothbeans insight



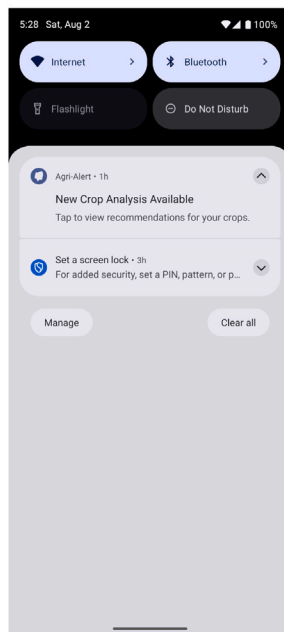
(c) News



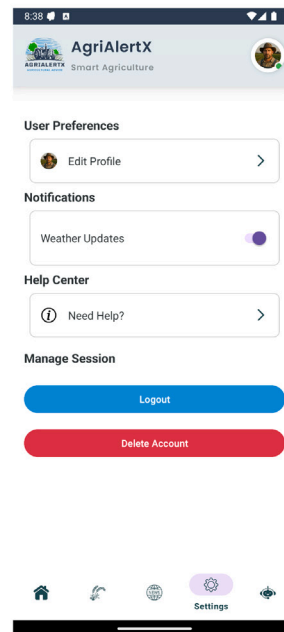
(d) ChatBot



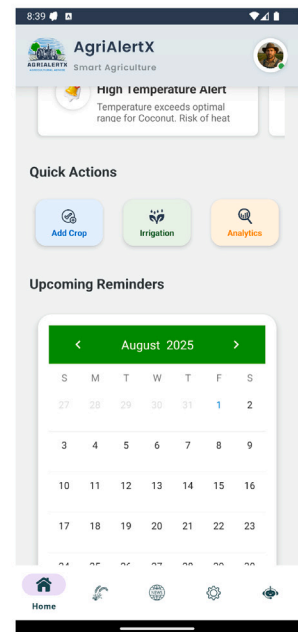
(e) Location Map



(f) Notification



(g) Settings



(h) Other features

Fig. 6. AgriAlertX android application: Additional features.

Table 3
Quality assurance results for different components.

Metric	Back-end	Web Front-end	Android App	iOS App
Quality Gate Status	Passed	Passed	Passed	Passed
Reliability Rating	A (0 issues)	D (23 issues)	A (4 issues)	A (0 issues)
Security Rating	A	A	A	A
Maintainability Rating	A (25 issues)	A (141 issues)	A (4 issues)	A (0 issues)
Code Duplication	1.6%	2.4%	1.2%	0.0%
Security Hotspots	1	2	2	0

Submissions will pass through a validation pipeline (completeness, consistency, geolocation, metadata conformity), with verified data used to recalibrate alert thresholds dynamically. The system ensures traceability, feedback integrity, and scientific rigor. Future versions may add a user-rating system and community dashboard to visualize aggregated contributions [34].

6. Quality assurance

A detailed quality assurance (QA) assessment was conducted in the back-end, web front-end, and mobile applications (Android and iOS) using SonarQube. This evaluation focused on essential software quality metrics, including *Reliability*, *Security*, *Maintainability*, *Code Duplication*, and *Code Coverage*. The summary of the results is presented in Table 3.

The security assessment showed excellent results across all components, each receiving an A rating and demonstrating strong adherence to best practices with no detected vulnerabilities. Maintainability was also rated A for the back-end and mobile apps, while the web front-end revealed concerns, with 141 issues and a D rating; resolving these is essential to reduce technical debt. Reliability assessments mirrored this trend: back-end, Android, and iOS scored A with no open issues, but the web front-end received a D due to 23 unresolved issues, highlighting the need for improved stability.

To address these web front-end issues and low test coverage, a refactor using modular design patterns is underway, alongside implementation of Jest and Cypress to raise code coverage to at least 80%. This will enhance maintainability, reduce technical debt, and support platform scalability.

Code duplication analysis was positive, with the back-end at 1.6%, web front-end at 2.4%, and both mobile apps at 0.0%. However, a critical finding was low automated test coverage: only 28.9% for the back-end and 0.0% for all front-ends, indicating risk of undetected defects. Structured testing is thus a priority for improved reliability.

Despite these challenges, all components passed the *SonarQube Quality Gate*, meeting baseline software standards. Continued focus on web front-end reliability and automated testing is vital for future improvement.

Complementing software quality assessment, system performance and usability were evaluated through simulations, forecast data, and user feedback. The average alert latency was 2.3 min (SD: 0.8), RMSE for 24-h temperature forecasts was 1.8 °C (MAE: 1.2 °C), and RMSE for 72-h forecasts reached 3.1 °C. A usability study with 33 smallholder farmers yielded a System Usability Score (SUS) of 82.6 [35], reflecting high user satisfaction. These preliminary results support AgriAlertX's responsiveness, forecast quality, and usability; broader field validation is planned for future deployments.

7. Conclusions

AgriAlertX represents a promising advancement in agricultural risk management by combining real-time weather forecasting with crop-specific, actionable recommendations. The current implementation leverages open-source weather APIs to deliver forward-looking alerts and practical guidance tailored to predicted meteorological conditions. While the system currently operates using threshold-based alerts, it

exhibits strong potential for enhancement through the integration of machine learning models and advanced predictive analytics.

Key development priorities include expanding automated test coverage, enriching the crop-weather knowledge base, and refining the accuracy and specificity of climate-driven recommendations. These improvements would strengthen the platform's predictive power and responsiveness in diverse agricultural contexts.

In the face of escalating climate uncertainty, tools like AgriAlertX can play a crucial role within the broader ecosystem of agricultural decision-support systems. Its open-source architecture and multi-platform accessibility foster inclusivity, enabling widespread adoption and iterative co-development.

Future iterations of the platform will also benefit from collaborative validation strategies, involving farmers, agronomists, and researchers in a participatory feedback loop. This community-driven approach will support the dynamic refinement of thresholds and recommendations, making AgriAlertX not just a technical solution, but a living, adaptive system shaped by its users.

AgriAlertX exemplifies how integrated digital tools can operationalize climate adaptation strategies in agriculture. Future deployments and collaborative feedback loops will be key to scaling AgriAlertX as a globally adaptive, data-informed agricultural resilience tool.

CRedit authorship contribution statement

Dadi Soukaina: Writing – review & editing, Writing – original draft, Software, Conceptualization. **Lachgar Mohamed:** Writing – review & editing, Validation, Supervision, Software, Conceptualization. **Fattouhi Radwa:** Writing – original draft, Software, Methodology. **Douidy Sifeddine:** Writing – original draft, Software, Conceptualization. **Hamid Himech:** Supervision. **Kartit Ali:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Food and Agriculture Organization of the United Nations (FAO). The impact of disasters and crises on agriculture and food security. 2022, Available: <https://www.fao.org/documents/card/en/c/cb9990en/>.
- [2] Ricciardi Valerio, Ramankutty Navin, Mehrabi Zia, Jarvis Lovell, Chookolingo Bruno. How much of the world's food do smallholders produce? *Glob Food Secur* 2018;17:64–72.
- [3] World Bank. Climate smart agriculture: Building resilience to climate change. 2021, Available: <https://www.worldbank.org/en/topic/climate-smart-agriculture>.
- [4] Jain Harshita, Dhupper Renu, Shrivastava Anamika, Kumar Deepak, Kumari Maya. Leveraging machine learning algorithms for improved disaster preparedness and response through accurate weather pattern and natural disaster prediction. *Front Environ Sci* 2022. Available: <https://www.frontiersin.org/articles/10.3389/fenvs.2023.1194918/full>.
- [5] Intergovernmental Panel on Climate Change (IPCC). Climate change 2022: Impacts, adaptation and vulnerability. 2022, Available: <https://www.ipcc.ch/report/ar6/wg2/>.
- [6] Docker Inc. Docker documentation. 2025, <https://www.docker.com/>.
- [7] Spring Team. Spring boot documentation. 2025, <https://spring.io/projects/spring-boot>.

- [8] Spring Team. Spring security documentation. 2025, <https://spring.io/projects/spring-security>.
- [9] Hibernate Team. Hibernate ORM documentation. 2025, <https://hibernate.org/orm/>.
- [10] Oracle Corporation. Mysql documentation. 2025, <https://www.mysql.com/>.
- [11] Pallets Projects. Flask web framework. 2025, <https://flask.palletsprojects.com/>.
- [12] Hugging Face. Hugging face transformers documentation. 2025, <https://huggingface.co/>.
- [13] Pedregosa Fabian, Varoquaux Gaël, Gramfort Alexandre, Michel Vincent, Thirion Bertrand, Grisel Olivier, Blondel Mathieu, Prettenhofer Peter, Weiss Ron, Dubourg Vincent, Vanderplas Jake, Passos Alexandre, Cournapeau David, Brucher Matthieu, Perrot Matthieu, Duchesnay Édouard, Ramírez Sebastián, et al. Python for crops analysis services: Scikit-learn and FastAPI. 2025, Scikit-Learn: <https://scikit-learn.org/>, FastAPI: <https://fastapi.tiangolo.com/>.
- [14] Vercel. Next.js documentation. 2025, <https://nextjs.org/>.
- [15] JetBrains. Kotlin programming language. 2025, <https://kotlinlang.org/>.
- [16] IncSquare. Retrofit for android. 2025, <https://square.github.io/retrofit/>.
- [17] Apple Inc. Swift programming language. 2025, <https://swift.org/>.
- [18] Alamofire Foundation. Alamofire for iOS. 2025, <https://github.com/Alamofire/Alamofire>.
- [19] Axios Authors. Axios: Promise-based HTTP client. 2025, <https://axios-http.com/>.
- [20] MDN Web Docs. Fetch API documentation. 2025, https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API.
- [21] SonarSource. SonarQube documentation. 2025, <https://www.sonarsource.com/sonarqube/>.
- [22] Open-Meteo Founders. Open-meteo. 2022-2025, <https://open-meteo.com>.
- [23] World News API Founders. World News API. <https://worldnewsapi.com>.
- [24] Food and Agriculture Organization of the United Nations (FAO). FAO crop information database. 2025, Available: <https://www.fao.org/land-water/databases-and-software/crop-information/en/>.
- [25] Food and Agriculture Organization of the United Nations (FAO). Ecocrop database: Crop-climate matching. 2025, Available: <http://ecocrop.fao.org/ecocrop/srv/en/home>.
- [26] United States Department of Agriculture (USDA). USDA plant hardiness zone map. 2025, Available: <https://planthardiness.ars.usda.gov/>.
- [27] Luo Qunying. Temperature thresholds and crop production: a review. *Clim Change* 2011;109(3):583–98.
- [28] Yang G, Liu J, Zhao C. Advancements in precision agriculture using AI. *Springer J Agric Sci* 2024;15:120–35.
- [29] Smith J, Brown L. Climate adaptation strategies for sustainable farming. *MDPI Clim Res* 2023;11:50–65.
- [30] Doe J, Roe R. Precision agriculture and AI: A review. *IEEE Trans Smart Agric* 2023;8:210–25.
- [31] Lee K, Park S. Building resilience in agriculture with AI. *J Agric Resil* 2024;9:90–105.
- [32] Garcia M, Patel A. Sustainable farming through AI-driven insights. *MDPI Sustain Agric* 2024;18:300–20.
- [33] Kumar S, Wang Y. Policy frameworks for AI in agriculture. *Springer J Agric Policy* 2024;7:45–60.
- [34] Wiggins Andrea, Crowston Kevin. Crowdsourcing: A tool for research data collection. *Comput Support Coop Work (CSCW)* 2011;20(1):1–28.
- [35] Brooke John. SUS: A quick and dirty usability scale. *Usability Eval Ind* 1996;189:4–7.