

# AgricultureEarlyWarning: A Web-Based Climate Advisory and Early Warning Platform for Albanian Farmers

Irida Gjermeni

Faculty of Economy and Agribusiness-Department of Mathematics and Informatics,  
Agricultural University of Tirana, Tirana, Albania

**Abstract**—This study reports the design and implementation of AgricultureEarlyWarning, a web-based prototype developed to operationalize climate information services and early warning logic for farmer-centered agricultural risk management in Albania. The system integrates parcel-level registration and geolocation, daily and hourly forecast ingestion, crop-stage-sensitive risk evaluation, satellite-derived indicators, dashboard analytics, AI-assisted agronomic explanation, vulnerability profiling, escalation support, and scheduled alert dissemination within a single application. The implementation stack combines ASP.NET MVC on .NET Framework 4.8, SQL Server persistence, Hangfire background scheduling, Open-Meteo weather services, Sentinel Hub APIs, and OpenStreetMap reverse geocoding. Rather than proposing a novel forecasting algorithm or claiming a complete national early warning system, the study contributes an implementation-oriented platform that links data acquisition, contextual risk interpretation, farmer-facing visualization, and asynchronous communication. The findings indicate that the prototype supports localized monitoring, advisory generation, alert scheduling, and analytics, while field validation, long-term adoption assessment, and full institutional response governance remain priorities for future work.

**Keywords**—AgricultureEarlyWarning system; climate information services; farmer advisory; remote sensing; ASP.NET MVC; Albania; decision support

## I. INTRODUCTION

Climate variability, extreme weather, and crop-specific sensitivity to temperature, rainfall, humidity, and wind continue to challenge agricultural production and farm-level decision-making. In this context, early warning systems (EWSs) are expected to do more than produce forecasts. A modern EWS is an integrated system that combines hazard monitoring and forecasting, risk assessment, communication, and preparedness processes so that people can take timely action before harmful events occur [1]. International frameworks for multi-hazard and people-centered EWSs emphasize four interconnected pillars: risk knowledge, detection and monitoring, warning dissemination and communication, and preparedness to respond [2], [3].

In agriculture, the practical value of climate services depends not only on the scientific quality of the forecast, but also on how well the information is translated into usable decisions. Previous work has shown that climate-service design must be aligned with user needs, decision timescales, and communication channels [4]. In smallholder and mixed farming contexts, mobile delivery channels such as SMS, phone-based interfaces, and localized advisories are

particularly important, but their effectiveness still depends on trust, relevance, language, and the capacity of farmers to act on the received information [5]. This means that agricultural EWS design should be farmer-facing, action-oriented, and operationally sustainable.

At the same time, recent literature shows growing interest in agricultural platforms that integrate fine-scale weather information, crop or hazard thresholds, mobile interfaces, digital advisory services, and risk-reduction workflows. Examples include farm-specific agrometeorological warning systems, nationwide agricultural disaster prevention architectures, forecast-driven disease warning systems, and digital advisory initiatives for smallholder agriculture [6]-[9], [19], [20]. These studies reinforce a common lesson: useful agricultural EWSs must connect scientific data, local context, and usable delivery mechanisms.

This study presents AgricultureEarlyWarning, a web-based application developed as a practical prototype for Albanian agriculture. The platform was designed to bridge climate information services (CIS) and agricultural early warning workflows within a single environment. Rather than functioning only as a weather dashboard, the application combines parcel registration, forecast retrieval, crop-aware risk interpretation, climate analytics, AI-assisted agronomic guidance, and scheduled dissemination of daily and event-based alerts. The interface is designed for farmer use, while the architecture also supports administrative oversight and future institutional escalation modules.

The contribution of this study is threefold. First, it documents the architecture and implementation logic of a deployable farmer-oriented agriculture app that integrates multiple open and commercial-ready services. Second, it maps the prototype explicitly to the core pillars of effective EWS design, allowing the reader to assess which components are already operational and which still require institutional completion. Third, it provides a realistic discussion of prototype maturity, including comparison with related agricultural EWSs, prototype-level functional validation, and current limitations. The study is, therefore, framed as a systems design and implementation study, suitable for practitioners and researchers interested in digital agriculture, climate services, and operational EWS development.

The remainder of the study is organized as follows: Section II reviews related work and motivates the design rationale. Section III explains the system requirements and EWS alignment. Section IV presents the architecture. Section V describes the prototype implementation. Section VI

discusses the functional evaluation and practical implications. Section VII outlines limitations and future work, and Section VIII concludes the study.

## II. RELATED WORK AND DESIGN RATIONALE

Climate services research consistently shows that information alone is insufficient if it is not embedded in a usable decision context. Born et al. demonstrated through a global meta-analysis that agricultural climate-service use varies by decision type, forecast horizon, and variable, highlighting the importance of matching information products to real farmer decisions [4]. Charalampopoulos and Droulia further stress that climate services for agriculture should move toward integrated, user-oriented pathways that link data generation to sector-specific application needs [6]. These arguments are directly relevant for agricultural apps, where interface simplicity and actionability can determine whether information is adopted or ignored.

Communication design is equally important. Yegbeme and Egah found that mobile-based dissemination is widely used in climate-service delivery to smallholder farmers, but relevance depends on socio-economic fit, literacy, trust in the information source, and the ability of farmers to interpret and respond to messages [5]. This literature supports the use of concise alerts, localized advice, and interfaces in the user's working language. For AgricultureEarlyWarning, this insight motivated the use of Albanian-language dashboards and SMS-oriented dissemination logic.

Recent agricultural EWS implementations also provide useful architectural cues. Shin et al. describe a farm-specific early warning service customized for farmers and extension workers, emphasizing cloud-based coordination, farm-level information delivery, and notification workflows [7]. Yao et al.

present a broader agricultural disaster prevention system that integrates forecast services, crop-specific warning logic, and information platforms for farmers and public actors [8]. Krupnik et al. show that a weather-driven wheat blast EWS achieved practical scale only because forecast models, user-centered interfaces, dissemination channels, and institutional cooperation were co-developed [9]. Together, these studies suggest that effective agricultural EWSs require four qualities: localization, interpretability, dissemination, and operational governance.

AgricultureEarlyWarning was designed in response to these lessons and to the needs of a context where farmers may require one integrated entry point for weather, risk, and agronomic interpretation. The design rationale, therefore, prioritizes: 1) parcel-specific localization rather than generic regional weather pages; 2) crop-aware risk interpretation rather than raw meteorological display; 3) actionable advisory text rather than only numerical outputs; 4) support for asynchronous daily and event-based dissemination; and 5) a modular architecture that can evolve from a farmer dashboard into a broader institutional EWS workflow.

Unlike many research prototypes that focus on a single hazard, crop, or model, AgricultureEarlyWarning aims to function as a flexible platform. It combines weather forecasts, selected remote-sensing indicators, crop-stage-sensitive logic, message scheduling, subscription management, and climate analytics in one web system. This broad scope increases implementation complexity, but it also reflects a real operational challenge: farmers do not experience data streams separately. They experience one decision environment in which weather, crop status, risk, timing, and advisory actions need to converge (see Table I).

TABLE I. COMPARATIVE POSITIONING OF AGRICULTUREEARLYWARNING AND ADVISORY SYSTEMS

System or study	Main focus	Localization	Dissemination	Validation or evidence	Position relative to AgricultureEarlyWarning
Shin et al. [7]	Farm-specific agrometeorological disaster warning service	Farm and extension-worker scale	Service notifications and cloud coordination	Operational service description	Stronger institutional service model; less emphasis on Albanian-language farmer dashboard and subscription workflow
Yao et al. [8]	Agricultural disaster prevention system and adaptation strategy	National/regional platform logic	Information platform for public and farmer actors	System and policy-oriented implementation	Broader governance scope; AgricultureEarlyWarning is lighter and more farmer-facing
Krupnik et al. [9]	Weather-driven wheat blast early warning system	Crop- and disease-specific local warnings	User-centered interfaces and dissemination	User-centered design, validation, and scaling	Stronger disease-specific validation; AgricultureEarlyWarning is multi-risk and modular but less field-validated
AgricultureEarly Warning	Web-based climate advisory and early-warning workflow for Albanian farmers	Parcel coordinates, crop association, vulnerability profile	Dashboard, daily advisory SMS, event-based alerts	Prototype-level functional validation and interface evidence	Combines CIS, parcel registration, risk logic, AI explanation, SMS, analytics, and escalation support in one modular prototype

## III. SYSTEM REQUIREMENTS AND ALIGNMENT WITH EARLY WARNING SYSTEM PRINCIPLES

The conceptual starting point of AgricultureEarlyWarning is that an agriculture app should not be evaluated only as a software product, but as a partial operational EWS. From this perspective, the system requirements were derived from the four core EWS pillars [1]-[3].

The first requirement is risk knowledge. The application must represent the local production context in a structured way. For this reason, the system begins with farmer registration and map-based parcel localization. Each farmer is associated with a geographic position, crop information, and cultivation context. This creates the basis for place-specific forecast retrieval and crop-sensitive interpretation. Risk knowledge is also supported by crop parameter definitions, stage-sensitive thresholds, risk

categories, and historical analytics that help transform environmental variables into meaningful agricultural signals.

The second requirement is detection, monitoring, analysis, and forecasting. The platform, therefore, integrates short- to medium-range weather forecast data and selected Earth observation products. Weather variables such as temperature, humidity, rainfall, wind speed, and precipitation probability are retrieved through Open-Meteo APIs [10]. Satellite-related indicators are designed through Sentinel Hub services, where the Processing API can provide imagery and the Statistical API can provide pixel-level summaries such as vegetation and moisture-related indicators without requiring full raster download [11]-[13]. These data sources are combined with rule-based crop logic to derive risk levels and advisory cues.

The third requirement is warning dissemination and communication. The application implements this by coupling dashboards with scheduled and event-driven messaging. Warnings must be timely, understandable, and actionable, not merely visible inside a control panel. This is why the platform supports daily advisory dissemination and repeated background checks for event-based alerts. Message content is designed to summarize the relevant risk, affected crop context, and a practical recommendation. This design is consistent with literature showing the importance of suitable delivery channels and understandable communication for smallholder-oriented climate services [5].

The fourth requirement is preparedness and response capability. This is the area where the current prototype is only partially complete. The farmer-facing side already supports a response-oriented workflow by translating risk scores into suggested actions. However, a full institutional response chain requires escalation management, acknowledgement workflows, case tracking, and formal roles for agronomists or administrators. AgricultureEarlyWarning includes the architectural direction for such modules, but not all of them are fully operational in the current implementation. This distinction is important because many digital platforms overclaim EWS completeness when they mainly cover monitoring and communication. In this study, preparedness is treated as an area of ongoing extension rather than a finished component.

A further requirement concerns sustainability and usability. The system is designed as a web application so that it can operate through common browsers and mobile-responsive interfaces. In addition, it includes a free trial and subscription workflow intended to support a realistic service model for climate information delivery. While economic adoption is outside the scope of the present systems paper, service sustainability is relevant because an EWS that cannot be maintained operationally will not achieve long-term value.

Finally, the system requires trustworthy geospatial localization. Reverse geocoding through Nominatim is used to transform coordinates into readable place references, while respecting API usage requirements and application identification rules [14], [15]. This supports clearer farmer records, contextual dashboards, and location-aware interpretation. Fig. 1 and Table II provide a compact visual interpretation of how these functions relate to the four pillars of an effective early warning system.

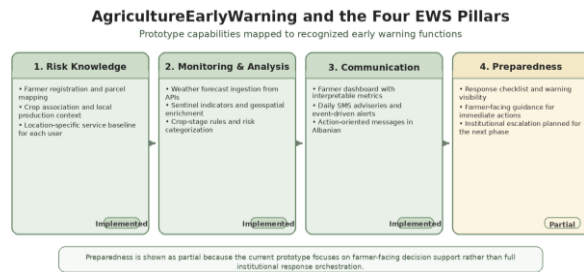


Fig. 1. Alignment of agricultureearlywarning capabilities with the four pillars of an effective early warning system.

TABLE II. ALIGNMENT OF AGRICULTUREEARLYWARNING WITH EWS PILLARS.

EWS pillar	Current support in prototype	Maturity
Risk knowledge	Parcel registration, crop association, crop parameters, location context, historical analytics	Strong
Detection and monitoring	Forecast ingestion, hourly/daily variables, satellite indicators, threshold logic	Strong
Dissemination and communication	Dashboard, daily advisory messages, event-based alerts, Albanian-language UI	Strong
Preparedness and response	Action suggestions available; institutional escalation and case workflows only partially implemented	Partial

#### IV. SYSTEM ARCHITECTURE

The overall architecture of AgricultureEarlyWarning is organized into five logical layers: data acquisition, geospatial and farmer context, risk and intelligence processing, application services, and dissemination. Fig. 2 summarizes this structure.

At the data-acquisition layer, the platform retrieves forecast and historical weather variables from Open-Meteo [10]. It is designed to request daily and hourly variables relevant to agriculture, including maximum and minimum temperature, relative humidity, rainfall, precipitation probability, and wind speed. In parallel, satellite-oriented services are integrated through Sentinel Hub APIs [11]-[13]. Sentinel-2 derived vegetation information, such as NDVI, contributes to crop condition interpretation, while Sentinel-1 derived signals such as VV and VH are used as proxies for moisture and crop structural conditions where appropriate. The goal is not to replace agronomic field observations, but to enrich the evidence available for localized advisory logic.

The geospatial and farmer-context layer stores farmer identity, parcel coordinates, crop associations, and administrative location fields. Parcel position is central because all downstream analysis is location-specific. Reverse geocoding adds readable place labels to coordinate-based records [14]. This layer also stores subscription state, trial status, and basic engagement-related attributes needed for service administration.

Above this layer sits the application-services layer, implemented in ASP.NET MVC on .NET Framework 4.8 [18] with SQL Server persistence and background scheduling via Hangfire [16], [17]. This layer exposes the farmer dashboard, registration workflows, climate analytics pages, administrative

controls, subscription handling, escalation support, and AI-assisted chat interface. Background jobs are used for recurrent weather checks and scheduled dissemination. The use of persistent job scheduling is important because warning systems must continue routine monitoring independently of individual user sessions.

The dissemination layer delivers outputs to end users. The primary interface is the responsive web dashboard, which provides current conditions, forecast trajectories, risk summaries, analytics, and recommendations. A second dissemination path uses SMS-style notifications for daily information and event-based alerts. This separation is intentional: a dashboard supports exploration, while alerts support timely action. In operational EWS terms, the dashboard corresponds more to situational awareness, whereas alerts correspond to communication and triggering.

The architecture was deliberately designed to remain modular. Each upstream service can be improved or replaced without rewriting the entire platform. For example, forecast providers can change, satellite workflows can be expanded, and the advisory-generation component can evolve from rule-based logic toward hybrid AI-assisted interpretation. This modularity is essential for deployment in real settings, where data availability, cost, and infrastructure constraints may change over time. The logical relationships between layers are visualized in Fig. 2.

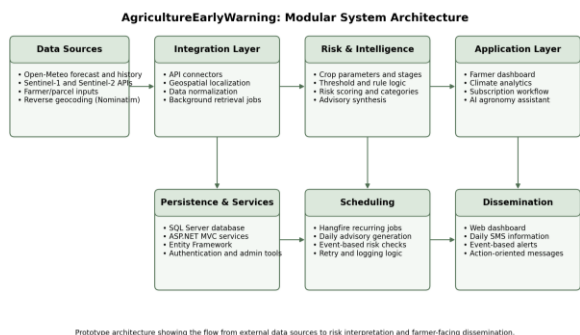


Fig. 2. Modular architecture of the AgricultureEarlyWarning prototype.

## V. PROTOTYPE IMPLEMENTATION AND INTERFACE REALIZATION

The current implementation of AgricultureEarlyWarning is realized as an ASP.NET MVC application on .NET Framework 4.8 with SQL Server persistence and Hangfire-based background processing. This section documents the main functional modules and provides interface-level evidence of how the prototype operationalizes registration, monitoring, analytics, communication, and preparedness-support functions. To improve technical transparency, the implementation is described at four levels: database structure, risk scoring logic, alert rules, and API integration (see Table III).

At the database level, the prototype uses normalized SQL Server entities to connect users, parcels, crops, warning records, SMS logs, vulnerability profiles, escalation events, and feedback. The database design separates static registration data from time-varying climate-risk records so that monitoring results can be audited and reused in analytics.

TABLE III. TECHNICAL IMPLEMENTATION DETAILS OF THE PROTOTYPE

Layer	Main implementation elements	Operational role
Database structure	Farmers, Crops, FarmerCrops, ClimateRiskPredictions, Sms Alerts, FarmerVulnerability Profiles, EwsAlertRecords, EwsEscalationEvents, Ews DispatchLogs, EwsFeedbacks, and EwsInstitutionalCases	Stores parcel context, crop associations, computed warnings, communication history, vulnerability data, escalation status, and feedback.
Risk scoring logic	Rule-based thresholds for temperature, rainfall, humidity, wind, crop stage, vulnerability score, and advisory category	Converts forecast variables into a 0-100 risk score and qualitative warning levels such as green, yellow, orange, and red.
Alert rules	Daily advisory job, event-based checks, acknowledgement status, escalation delay, and retry/logging logic	Supports asynchronous warning dissemination and prepares the system for institutional escalation workflows.
API integration	Open-Meteo forecast requests, Sentinel Hub Processing/Statistical APIs, Nominatim reverse geocoding, and Hangfire scheduling services	Combines external data acquisition with geospatial localization, remote-sensing enrichment, and recurring background execution.
AI explanation layer	Server-side AIChatService with constrained prompt template and deterministic risk facts supplied by the warning engine	Explains warnings and general management actions without changing the computed risk level or acting as the trigger mechanism.

The risk score is calculated as an interpretable engineering score rather than as a learned statistical forecast. Each relevant meteorological variable is normalized against crop- and stage-specific thresholds. The resulting partial scores are multiplied by crop-stage weights and combined with vulnerability modifiers. The final score is mapped into warning levels so that the farmer sees a simple risk category while the administrator can still inspect the underlying variables.

Alert rules are implemented as scheduled and event-triggered routines. The daily advisory routine summarizes expected conditions for the active parcel and crop, while event-based checks are used when current or forecast conditions exceed configured thresholds. The escalation rule screen defines acknowledgement windows and transfer points from farmer notification to agronomist or administrator review. In the current prototype, these workflows support preparedness logic, but formal institutional operating procedures are still required before the platform can be described as a complete institutional EWS.

API integration is isolated behind service classes so that external providers can be changed without rewriting the dashboard or alert modules. Open-Meteo supplies daily and hourly forecast variables [10]. Sentinel Hub services support satellite-derived indicators and statistics [11]-[13]. Nominatim is used for reverse geocoding and readable location labels [14], [15]. Hangfire executes recurring checks and scheduled dissemination independently of user login sessions [16], [17].

### A. Farmer Registration and Localization

The onboarding flow allows a farmer to register and select a parcel on the map. Geographic coordinates are stored

directly and then enriched with reverse-geocoded location information. This step is foundational because subsequent monitoring, forecast requests, and risk interpretation are all tied to the selected parcel. The registration process is also connected to free-trial and subscription logic, reflecting the practical requirement that operational climate information services need a sustainable delivery model.

### B. Weather and Advisory Dashboard

After registration, the farmer accesses a dashboard that combines forecast information, risk summaries, and practical guidance. The platform supports daily and hourly weather views and presents metrics in agricultural terms rather than as raw meteorological lists. This includes explicit units, intuitive risk badges, and concise explanatory text. The design seeks to reduce interpretation burden for the user while preserving transparency about the underlying variables.

### C. Crop-Aware Risk Logic

The risk engine evaluates conditions such as temperature extremes, excessive humidity, rainfall accumulation, and wind intensity relative to crop-stage sensitivity and configured crop thresholds. This allows the same forecast signal to be interpreted differently for different crop and stage combinations. For instance, a humidity and rainfall episode may be treated as a stronger disease risk during sensitive crop stages than during less vulnerable periods. The platform, therefore, behaves as a decision-support layer on top of the forecast rather than as a forecast viewer only.

### D. Remote-Sensing Enrichment

The architecture supports the use of Sentinel-derived indicators to complement weather-based interpretation. NDVI can be used as a proxy for vegetation vigor, while Sentinel-1 backscatter-derived values can contribute to moisture and structural interpretation. In the current prototype, these indicators are integrated as additional evidence streams that can enrich dashboards and analytics. Their role is supportive rather than fully autonomous, since agronomic interpretation still benefits from local calibration and field knowledge.

### E. AI-Assisted Agronomic Guidance

The system contains an AI-assisted chat component implemented as a server-side AIChatService connected to a configurable large-language-model endpoint. The component is not used to calculate risk scores or trigger alerts. Instead, it receives structured facts from the deterministic warning engine, including crop, stage, location context, relevant weather variables, computed risk level, and the active advisory category. The prompt template constrains the model to explain why the warning appears, translate technical variables into farmer-friendly Albanian, and provide conservative general actions. It also instructs the model not to invent sensor values, not to change the computed risk level, and not to provide pesticide doses unless such advice is grounded in an official label or validated agronomic source. This design follows the emerging view that generative AI can support agricultural advisory communication, but that guardrails, domain facts, and human or expert oversight are necessary for safe deployment [21], [22]. Generated text is, therefore, treated as

an interpretability layer, while the operational warning logic remains deterministic and auditable.

### F. Background Scheduling and Notification

Routine monitoring and dissemination are handled through Hangfire recurring jobs [16], [17]. This allows the application to execute regular checks, such as periodic weather retrieval and event-driven warning evaluation, as well as scheduled daily information delivery. From an EWS perspective, this is a critical operational feature because alerts should not depend on a farmer being logged into the platform at the moment a hazardous condition develops.

### G. Climate Analytics and Administration

Beyond immediate alerts, the prototype supports climate analytics views that aggregate historical records, summary metrics, and time series. This supports reflection and learning, not just immediate action. Administrative features allow oversight of farmer records, crop associations, and other service-management functions. In future iterations, these administrative capabilities are expected to evolve into institutional response and case-management modules.

Fig. 3 synthesizes the implemented operational chain from onboarding and data retrieval to advisory delivery and learning-oriented analytics. Table IV summarizes the main implemented modules, while Fig. 4(a) to Fig. 4(d) and Fig. 5(a) to Fig. 5(d) provide enlarged interface-level evidence from the deployed prototype.

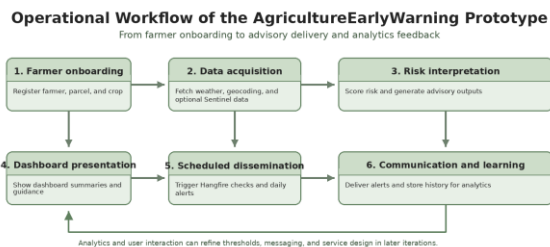


Fig. 3. Farmer-facing operational workflow of the AgricultureEarlyWarning prototype from onboarding to advisory dissemination and analytics feedback.

TABLE IV. MAIN IMPLEMENTED MODULES OF THE PROTOTYPE

Module	Implemented function	Operational note
Farmer onboarding	Registration, parcel geolocation, crop selection, subscription/trial linkage	Creates parcel-specific service context
Weather dashboard	Current, hourly, and daily forecast display with agricultural units and risk badges	Farmer-facing situational awareness
Risk engine	Crop-stage-sensitive thresholds for temperature, humidity, rainfall, and wind	Transforms forecast values into warnings
Remote-sensing enrichment	Sentinel-1 / Sentinel-2 indicator support including NDVI and backscatter summaries	Complements weather-based interpretation
AI guidance	Conversational explanation layer in Albanian	Improves interpretability, not core warning logic
Dissemination	Daily advisory SMS and event-based alerts via recurring jobs	Supports asynchronous communication
Climate analytics	Historical summaries, trends, and export-oriented views	Supports review and learning

H. Interface-Level Implementation Evidence

Fig. 4(a) to Fig. 4(d) and Fig. 5(a) to Fig. 5(d) present enlarged screenshots from the realized prototype. They document how the implementation translates the architectural

workflow into concrete user interfaces for parcel geolocation, dashboard monitoring, analytical interpretation, vulnerability assessment, survey capture, escalation management, and farmer communication through SMS-style dissemination.

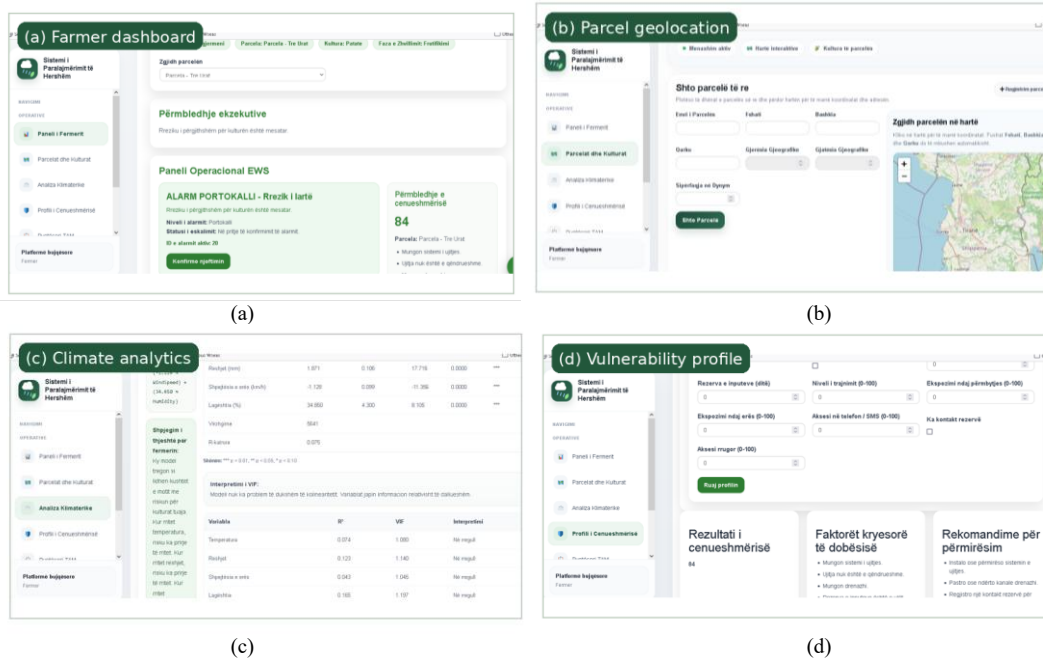


Fig. 4. (a) Farmer dashboard showing parcel selection, current risk summary, acknowledgement status, and vulnerability score. (b) Parcel geolocation screen where the farmer enters parcel attributes and selects coordinates on the interactive map. (c) Climate analytics screen with regression-style summary indicators, VIF interpretation, and climate variable diagnostics. (d) Vulnerability profile screen recording, exposure, access, training, backup capacity, and practical recommendations.

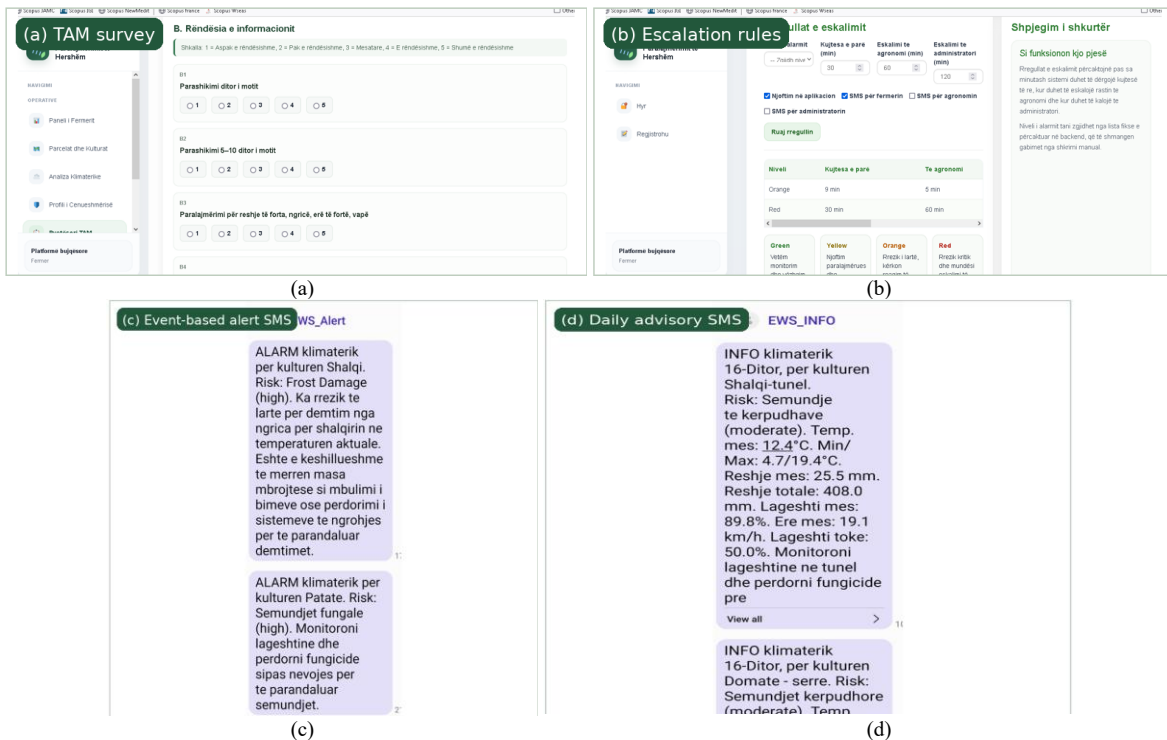


Fig. 5. (a) TAM survey capture screen used to collect structured user perceptions about the advisory service. (b) Escalation-rule configuration screen defining acknowledgement delays, transfer to agronomist, and transfer to administrator. (c) Event-based alert SMS showing risk type, severity, crop context, and immediate recommended action. (d) Daily advisory SMS showing forecast summary, disease-related risk, wind, rainfall, humidity, and suggested monitoring action.

## VI. FUNCTIONAL EVALUATION AND TECHNICAL DISCUSSION

This study does not report a randomized field trial or a statistically validated impact assessment. Instead, the revised evaluation reports prototype-level functional validation, architectural coherence, and implementation evidence drawn from the realized workflow and user interfaces. This evaluation mode is appropriate for a design and implementation paper, but it is clearly separated from field effectiveness, adoption, or yield-impact claims.

The functional validation focuses on measurable implementation indicators rather than agronomic impact. Table V summarizes the verification evidence available at prototype stage. The table deliberately distinguishes implemented functionality from performance metrics that require future field deployment, such as response latency under production load, API uptime over a full season, and farmer acceptance after repeated use.

TABLE V. PROTOTYPE-LEVEL FUNCTIONAL VALIDATION INDICATORS

Validation dimension	Measured or verifiable indicator	Prototype evidence	Remaining gap
EWS pillar coverage	Four EWS pillars reviewed; three assessed as strong and one as partial	Table II maps risk knowledge, monitoring, dissemination, and preparedness	Preparedness requires formal institutional operating procedures.
Functional module coverage	Seven implemented module groups plus escalation, feedback, and vulnerability-support screens	Table IV and Fig. 4(a)-Fig. 5(d) document the workflow	Independent usability testing is still needed.
Scheduler configuration	Recurring event checks and daily advisory dissemination configured through Hangfire	Daily EWS_INFO and event-based EWS_ALERT examples are captured in Fig. 5c and Fig. 5d	Long-term scheduler accuracy and latency must be measured under production load.
Risk-output transparency	Risk level, crop context, weather variables, vulnerability score, and advisory text stored or displayed	Dashboard, analytics, SMS, and vulnerability screens expose the key decision variables	Threshold calibration across seasons and crops remains future work.
API integration coverage	Weather, satellite, reverse geocoding, persistence, and background-job services integrated	Open-Meteo, Sentinel Hub, Nominatim, SQL Server, and Hangfire are separated through service classes	API reliability and rate-limit behavior require season-long monitoring.
Communication outputs	Two communication formats: daily advisory and event-based alert	SMS-style messages include crop, risk type, severity, and action-oriented text	Message comprehension and trust should be tested with farmers and extension staff.

The first strength of the prototype is localization. By structuring the workflow around parcel coordinates and crop associations, the system avoids the genericity of broad regional weather dashboards. This is consistent with recent agricultural EWS examples that emphasize farm-specific or location-specific service logic [7]-[9]. The use of remote-sensing services further extends the potential for localized interpretation, especially in settings where in-field instrumentation is limited.

The second strength is dissemination design. The system recognizes that warnings must reach users through appropriate channels and at suitable times. This reflects prior evidence that communication channels are central to climate-service usability for farmers [5]. By combining an exploratory dashboard with daily and event-based alerts, the platform balances information richness with timely communication. The Albanian-language interface also supports contextual usability.

The prototype also has several limitations. First, the agronomic quality of risk outputs depends on the calibration of crop parameters, thresholds, and advisory texts. Without extended field validation across crops, seasons, and agroecological zones, warning quality should be interpreted cautiously. Second, the integration of satellite indicators is promising but still requires rigorous calibration and validation for operational agricultural inference. Third, the platform currently emphasizes the farmer-facing and analytical sides of the EWS more than the institutional response side. It is therefore presented as a farmer-centered prototype with partial preparedness support, not as a complete national or institutional EWS.

The fourth limitation concerns operational infrastructure. Because the system depends on external APIs, background processing, and hosting reliability, production deployment requires attention to service continuity, usage policies, rate limits, scheduler robustness, and data-protection rules [10]-[17]. This is not a conceptual weakness of the design, but it is a real implementation constraint that any operational EWS must manage.

The generalizability of the architecture is strongest where three assumptions hold: geospatial parcel coordinates can be collected, weather and satellite APIs are accessible, and an extension or administrative actor can respond to escalated warnings. The software pattern is transferable because the data-acquisition, risk-logic, dashboard, and dissemination layers are modular. However, thresholds, language, crops, institutions, SMS providers, and payment models must be localized before deployment in another region or country. The platform is therefore best understood as a reusable architectural blueprint rather than a directly universal solution.

## VII. LIMITATIONS AND FUTURE WORK

Several development directions emerge from the current prototype and from the limits of the functional validation reported above.

First, field validation should become the priority. Future work should compare risk outputs and advisories with observed crop conditions, farmer actions, and actual hazard

outcomes across seasons. This would enable calibration of thresholds and the estimation of practical utility.

Second, the preparedness and response pillar should be expanded from prototype support into a formal institutional workflow. The current interface includes vulnerability profiling, escalation-rule configuration, feedback capture, and communication examples. Future deployment should add formal alert acknowledgement, agronomist assignment, dispatch logs, case closure rules, and institutional accountability procedures. Such additions would move the system from a strong farmer information platform toward a more complete end-to-end EWS.

Third, remote-sensing integration should be strengthened. Future work can evaluate how NDVI, VV, VH, and related indices improve risk prediction, especially for stress detection, excess moisture interpretation, and disease-supportive conditions. At present, these indicators are best understood as complementary evidence streams.

Fourth, the system would benefit from broader communication options and user testing. SMS remains important, but future versions may include WhatsApp-like channels, richer multilingual content, structured advisory templates, and usability testing with farmers and extension personnel.

Fifth, longer-term economic and adoption evaluation is needed. Since climate information services are also service products, future studies should assess farmer acceptance, repeated use, willingness to pay, and the institutional sustainability of subscription-based or mixed delivery models.

Finally, future technical work may include migration toward newer .NET stacks, stronger API abstraction, automatic performance logging, dashboard latency monitoring, improved audit trails, and richer analytics for both farmers and administrators. These enhancements would improve maintainability and long-term deployment readiness.

## VIII. CONCLUSION

This study presented AgricultureEarlyWarning as a design and implementation case of a web-based climate advisory and early warning platform for farmer-centered agricultural risk management in Albania. The prototype integrates parcel-based localization, weather and satellite data services, crop-aware risk interpretation, dashboard analytics, AI-assisted explanation, vulnerability profiling, escalation support, and scheduled dissemination inside a single operational environment. Its main contribution is the engineering of a modular workflow that links risk knowledge, monitoring, communication, and response support in a form that is implementable with existing web technologies and external data services.

The study shows that a practical agricultural EWS pathway does not need to start as a monolithic national infrastructure. It can begin as a modular, farmer-facing platform that incrementally connects risk knowledge, monitoring, warning communication, and response support. AgricultureEarlyWarning already demonstrates meaningful progress in the first three of these areas and provides prototype-

level support for the fourth. Institutional preparedness and coordinated response remain the next major step.

For digital agriculture researchers and practitioners, the platform illustrates how climate information services can be transformed into a more actionable service layer for farming decisions. For implementation contexts similar to Albania, it provides a realistic blueprint for integrating open weather services, satellite APIs, geospatial localization, and warning dissemination into one coherent system. The prototype, therefore, represents a substantive step toward practical, localized, and scalable agricultural early warning support.

## REFERENCES

- [1] UNDRR, "Early warning system," UNDRR Terminology, 2009. [Online]. Available: <https://www.undrr.org/terminology/early-warning-system>. Accessed: Apr. 17, 2026.
- [2] World Meteorological Organization, Multi-hazard Early Warning Systems: A Checklist. Geneva, Switzerland: WMO, 2018.
- [3] UNDRR, "Early Warnings for All," 2024. [Online]. Available: <https://www.undrr.org/implementing-sendai-framework/sendai-framework-action/early-warnings-for-all>. Accessed: Apr. 17, 2026.
- [4] L. Born, S. Prager, J. Ramirez-Villegas, and P. Imbach, "A global meta-analysis of climate services and decision-making in agriculture," *Climate Services*, vol. 22, p. 100231, 2021, doi: 10.1016/j.cliser.2021.100231.
- [5] R. N. Yegbemey and J. Egah, "Reaching out to smallholder farmers in developing countries with climate services: A literature review of current information delivery channels," *Climate Services*, vol. 23, p. 100253, 2021, doi: 10.1016/j.cliser.2021.100253.
- [6] I. Charalampopoulos and F. Droulia, "A pathway towards climate services for the agricultural sector," *Climate*, vol. 12, no. 2, p. 18, 2024, doi: 10.3390/cli12020018.
- [7] Y.-S. Shin, H.-A. Lee, S.-H. Park, Y.-K. Han, K.-M. Shim, and S.-J. Han, "Establishment and operation of an early warning service for agrometeorological disasters customized for farmers and extension workers at metropolitan-scale," *Atmosphere*, vol. 16, no. 3, p. 291, 2025, doi: 10.3390/atmos16030291.
- [8] M.-H. Yao, Y.-H. Hsu, T.-Y. Li, Y.-M. Chen, C.-T. Lu, C.-L. Chen, and P.-Y. Shih, "Agricultural disaster prevention system: Insights from Taiwan's adaptation strategies," *Atmosphere*, vol. 15, no. 5, p. 526, 2024, doi: 10.3390/atmos15050526.
- [9] T. J. Krupnik et al., "A weather-forecast driven early warning system for wheat blast disease: User-centered design, validation, and scaling in Bangladesh and Brazil," *Climate Services*, vol. 39, p. 100589, 2025, doi: 10.1016/j.cliser.2025.100589.
- [10] P. Zippenfenig, "Open-Meteo.com Weather API," Zenodo, 2023, doi: 10.5281/zenodo.7970649.
- [11] Sentinel Hub, "Sentinel Hub API documentation," Sentinel Hub Documentation, Sinergise Ltd., 2026. [Online]. Available: <https://docs.sentinel-hub.com/api/latest/>. Accessed: Apr. 17, 2026.
- [12] Sentinel Hub, "Processing API," Sentinel Hub Documentation, Sinergise Ltd., 2026. [Online]. Available: <https://docs.sentinel-hub.com/api/latest/api/process/>. Accessed: Apr. 17, 2026.
- [13] Sentinel Hub, "Statistical API," Sentinel Hub Documentation, Sinergise Ltd., 2026. [Online]. Available: <https://docs.sentinel-hub.com/api/latest/api/statistical/>. Accessed: Apr. 17, 2026.
- [14] Nominatim, "Reverse geocoding," Nominatim 5.3.1 Manual, OpenStreetMap Foundation, 2026. [Online]. Available: <https://nominatim.org/release-docs/latest/api/Reverse/>. Accessed: Apr. 17, 2026.
- [15] OpenStreetMap Foundation, "Nominatim Usage Policy," OpenStreetMap Operations, 2026. [Online]. Available: <https://operations.osmfoundation.org/policies/nominatim/>. Accessed: Apr. 17, 2026.
- [16] Hangfire, "Performing recurrent tasks," Hangfire Documentation, 2026. [Online]. Available: <https://docs.hangfire.io/en/latest/background-methods/performing-recurrent-tasks.html>. Accessed: Apr. 17, 2026.

- [17] Hangfire, "Making ASP.NET applications always running," Hangfire Documentation, 2026. [Online]. Available: <https://docs.hangfire.io/en/latest/deployment-to-production/making-aspnet-app-always-running.html>. Accessed: Apr. 17, 2026.
- [18] Microsoft, "ASP.NET MVC 5," Microsoft Learn, 2024. [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>. Accessed: Apr. 17, 2026.
- [19] Food and Agriculture Organization of the United Nations, The Impact of Disasters on Agriculture and Food Security 2025: Digital Solutions for Reducing Risks and Impacts. Rome, Italy: FAO, 2025.
- [20] AIM for Scale, Digital Advisory Services for Agriculture: Innovation Package. Washington, DC: Agricultural Innovation Mechanism for Scale, 2025.
- [21] AIEP Initiative, S. Collis, F. Kinyua, V. Kumar, H. Lakougna, C. Merz, K. Pandey, and C. Resch, Disrupting Agricultural Advisory Utilizing Generative AI: Lessons Learned from Minimum Viable Product Development of the Agricultural Information Exchange Platform. Gates Foundation, GIZ, and CLEAR Global, 2025.
- [22] S. Yang, Z. Yuan, S. Li, R. Peng, K. Liu, and P. Yang, "GPT-4 as evaluator: Evaluating large language models on pest management in agriculture," arXiv preprint arXiv:2403.11858, 2024.