Towards a Smart Farming Platform: From IoT-Based Crop Sensing to Data Analytics

Héctor Cadavid(✉), Wilmer Garzón, Alexander Pérez, Germán López, Cristian Mendivelso, and Carlos Ramírez

Escuela Colombiana de Ingeniería, Bogotá, Colombia
{hector.cadavid,wilmer.garzon,alexander.perez}@escuelaing.edu.co,
{german.lopez-p,cristian.mendivelso,
carlos.ramirez-ot}@mail.escuelaing.edu.co
http://www.escuelaing.edu.co

Abstract. Colombia is a country with a huge agricultural potential, thanks to its size and geography diversity. Unfortunately, it is far from using it efficiently: 65% of its farmland is either unused or underused due to political problems. Furthermore, vast of Colombian agriculture is characterized - when compared with other countries - by low levels of productivity, due to the lack of good farming practices and technologies.

The new political framework created by the recently signed peace agreement in this country opens new opportunities to increase its agricultural vocation. However, a lot of work is still required in this country to improve the synergy between academia, industry, agricultural experts, and farmers towards improving productivity in this field.

Advances in smart-farming technologies such as Remote Sensing (RS), Internet of Things (IoT), Big Data/Data Analytics and Geographic Information Systems (GIS), bring a great opportunity to contribute to such synergy. These technologies allow not only to collect and analyze data directly from the crops in real time, but to extract new knowledge from it. Furthermore, this new knowledge, combined with the knowledge of local experts, could become the core of future technical assistance and decision support systems tools for countries with a great variety of soils and tropical floors such as Colombia.

Motivated by these issues, this paper proposes an extension to Thingsboard, a popular open-source IoT platform. This extended version aims to be the core of a cloud-based Smart Farming platform that will concentrate sensors, a decision support system, and a configuration of remotely controlled and autonomous devices (e.g. water dispensers, rovers or drones). The architecture of the platform is described in detail and then showcased in a scenario with simulated sensors. In such scenario early warnings of an important plant pathogen in Colombia are generated by data analytics, and actions on third-party devices are dispatched in consequence.

Keywords: Smart farming · Data analytics · Precision agriculture IoT

J. E. Serrano C. and J. C. Martínez-Santos (Eds.): CCC 2018, CCIS 885, pp. 237–251, 2018.
https://doi.org/10.1007/978-3-319-98998-3_19

1 Introduction

By 2050, a world population of nearly 9.1 billion people has been estimated, which would require increasing overall food production by at least 70% (compared with 2007 production statistics) [5]. Given this unsettling scenario, it is not surprising that food security policies are among the main goals in a global agenda. However, to achieve this Policies, the amount of soil devoted to agriculture should be incremented and be more efficient. A higher amount of areas devoted to agriculture, with bad or outdated farming practices, lead not only to low productivity rates but to the increase of other problems like water contamination by excessive dosage of pesticides [7].

Colombia is a strategic case study for this problem. Despite of being a large country of 114 Mha (twice as Spain), with 42 Mha suitable for agriculture, five thermal floors, and a great diversity in terms of soil, geology, topography and vegetation, it is increasingly supplying its food needs through imports (by 2016, 30% of the food consumed by its population in one year). These statistics are explained by the fact that the country is using only an about of a third of the available agricultural land (14 Mha out of 42 Mha), according to the Rural Agricultural Planning Unit (UPRA, acronym in Spanish). In addition, most of such agriculture is characterized by low levels of technology, due to almost 50-years of internal conflict deterring investment in secluded farms.

However, the new political framework created by the recently signed peace deals open new opportunities to increase the agriculture vocation of the country. Indeed, the FAO has defined Colombia as one of the possible agricultural leaders for the world, and a key actor in the fight against hunger and malnutrition.

This paper describes the initial results of a research project whose final goal is to create a MaaS (Monitoring as a Service) platform that enables the synergy between IoT technology, Data-Analytics, and experts in Colombian agricultural species. This platform, which aims to be the core of future technologies for Colombian agriculture, is expected to enable a knowledge-feedback process as the one described below:

1. A set of soil sensors, distributed through several crops, transmits data (environment and soil variables) to the MaaS platform.
2. The MaaS platform, based on the rules for pre-known and pre-configured risks and threats, fires an alarm when the conditions are met.
3. When an alarm is fired, two additional actions could be performed: (1) a static actuator (e.g. an irrigation sprinkler) is remotely activated, or (2) a request for a precision-agriculture-task (e.g. applying a pesticide) is sent to the control center of an autonomous robot fleet [6].
4. When an anomaly (still not a risk) is identified, the system could also request (through the robot fleets control center) a data-gathering task, such as taking multi-spectral pictures through a Drone or a Rover.
5. An expert, as a daily basis routine, or motivated by the anomaly detection, checks all the data (sensor readings, pictures, crop's relative localization and history). The expert, based on such information, and further analysis -if

required- could register a spatial-temporal classification Tag (e.g. the name of a disease).
6. Once enough spatial-temporal Tags have been registered through the normal operation of the platform, a classifier (e.g. to identify the disease) is trained. Such classifier is then included as a component into the MaaS system so that future readings would allow the automatic detection of the newly identified disease.

The platform is built upon *Thingsboard* [19], a popular open-source IoT software for device management, data collection, processing and visualization. The extensions proposed in this paper for the Thingsboard architecture, so far, will include the functional requirements of the steps 1 through 4 of the scenario described above. As a study case, a simulated scenario for the early detection of the *Phytophthora infestans* [8] pathogen is described, which makes use of the following features provided by the new architecture:

– Extended data model with sensor/crops/farms detail level, and concepts from The International Center for Tropical Agriculture guidelines [2].
– API for accessing the extended data model from within the rules, and storing/accessing intermediate states of it.
– Geo-referenced data indexing and GIS capabilities.
– High-resolution photos storage and indexing.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the proposed architecture built upon the Thingsboard platform. Section 4 describes the problem of the *Phytophthora infestans*, how its early detection is addressed with the proposed platform, and the outcome of preliminary experiments. Section 5 concludes the paper.

2 Related Work

It is not surprising that the vast amount of research related to IoT and BigData applications in different fields (Ahmed et al. did a general survey in [1]), given the exponential growth in the data collected around the world (it has been said that up to 90% of the world's data has been produced after 2011 [12]). When it comes to applications in agriculture, and given all the factors that affects its productivity such as climate, soil, pests, diseases, and weather [15] there are two main approaches in previous work: (1) how to gather and transmit data from the crop and (2) how to process data and perform actions based on such processing outcomes. For the first approach, there is a complete survey of communications-related topics on wireless sensor network (WSN), such as communication protocols and energy efficiency in [11]. For the second approach, Lasso et al. [13] proposed the AgroCloud platform [14], which aims to the prevention of coffee rust. Although this platform generates early warnings based on data collected by third-party weather data providers, including air temperature, relative humidity, wind speed and direction, rain and solar radiation, it is not open for processing data transmitted directly from crop sensors. Verdouw et al. [21], proposed an analysis and decision-making model for a supply chain management in Dutch floriculture industry. Peisker et al. [17], describes a data-analysis model created by John Deere Company to keep trace of tractors' performance using big data and data collected from devices in the field. Although not specifically intended for agricultural applications, for this approach there are other works worth mentioning given its application of real-time data processing of environmental sensors. Bashir et al. [3] presented a framework for the analysis of large amounts of data from smart buildings, including oxygen levels, smoke/hazardous gases, luminosity, among others. Sarangi et al. [18] proposed framework for an agricultural advisory call center; here the farmer sends images of plants with crop diseases and the system makes the diagnosis and indicates the appropriate management of the disease. This framework does not allow real-time data processing, nor the detection of diseases from the information of the sensors.

One of the closest works to the proposal in this paper, when it comes to long-term objectives, is *FarmBeats* [20], a platform that covers both approaches. On the one hand, *FarmBeats* addressed the problem of how to transmit efficiently (in terms of energy and speed) data from sensors in regions with low coverage of communications infrastructure. In contrast, it proposed an architecture that aims to local data processing, integration with drones control (to take pictures) and cloud-based data persistence for centralized data-analytics. There are, however, two main differences with our proposal:

– FarmBeats is a complete, full-fledged technology, with a fixed set of hardware and software components. This platform, on the other hand, has a narrower scope as an extensible software platform, where new devices, new third-party systems, and, more importantly, new data-analytics strategies for early phenomenon detection (e.g. diseases) could be integrated with ease. Moreover, this extension for Thingsboard is expected to be accepted (*pulled*) by the community, and become the core of advanced smart-farming/IoT solutions with the integration of custom devices (sensors, actuators, and autonomous robots).
– FarmBeats processes data at two different places: at local PC, for sensor monitoring and decision making, and in the cloud, to perform cross-data analytics with the information provided by all the farms in conjunction. The architecture of our proposal, on the other hand, aims to be a centralized, cloud-based smart farming solution that performs data analytics and decision taking in one place. Thus, the scope of a decision-making tasks in our platform will not be limited to the local context of the event.

3 Proposed Extension Points

This section presents the main contribution of this paper. The key idea is to extend the *Thingsboard* platform from both the functional and architectonic point of views, in order to make it suitable for the application scenarios described in Sect. 1.

3.1 Thingsboard - Base Architecture and Data Model

The base architecture of Thingsboard aims to a high scalability through the distribution of its workload across multiple processing nodes without a single point of failure. Such workload distribution is achieved with the actors' model proposed by Hewitt et al. [9] and its implementation through the Akka platform [4].

Thingsboard was designed not only with scalability in mind, but also for front-end customization. On the one hand, its Widgets model enables the integration of new UI modules (for data visualization, alarms management, etc.). On the other hand, the Thingsboard Rule Engine allows to process messages from devices and trigger actions through plugins. One of the most useful plugins is the one provided to enable interoperability with Apache Spark, an analytics engine for large-scale data processing. Although a detailed description of Thingsboard architecture (at actors-level) is described on its official website[1], Fig. 1 presents an schema of the higher level components (and their interactions) which would be involved in a conventional IoT application case study.

As described in such figure, a conventional Thingsboard configuration is limited to the analysis of data collected from sensors in real time. This, as mentioned before, is a big limitation for applications scenarios such as the early detection of diseases in crops, whose rules might require access not only to real-time sensor readings, but to crop details and its historical information. Furthermore, if an advanced action for such rule is expected (e.g. an autonomous drone action), geo-referenced information would be required as well.

In order to enable adaptability when it comes to scenarios configuration, Thingsboard defines in its data and widget models the *Asset* entity. An *Asset* is an abstract IoT entity which could be related to other assets and devices (e.g. sensors) and therefore allowing a hierarchical composition of such devices. For example, a scenario of a farm with two crops, each one with two sensors, could be defined (directly through Thingsboard user interface) as a root asset (for the farm), two child assets (for the crops), each one with two devices. Furthermore, UI-widgets and dashboards for a hierarchical model as the aforementioned could be easily configured.

3.2 Proposed Extensions

Extended Data Model and Architecture. Although the *Asset* abstraction makes Thingsboard a highly flexible platform for most IoT application scenarios, it isn't enough to represent the information which is expected to be captured and processed (by data analytics techniques) in our scenarios. Our IoT/data-analytics application goals would require not only a devices hierarchy, but details such as crop history, application of good agricultural practices, geo-referenced information and pictures, among others, as described in Fig. 2.

[1] https://thingsboard.io/docs/reference/architecture/.

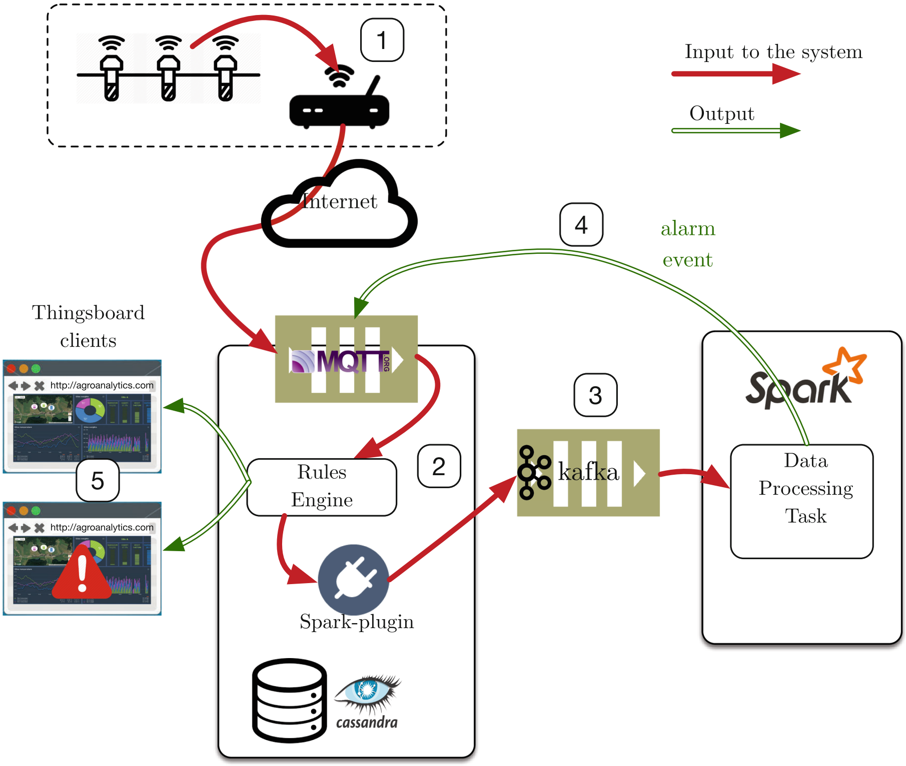


Fig. 1. (1) Sensors transmit data stream (through an Internet-gateway) using one of the protocols currently supported: MQTT, CoAP or HTTP. (2) Thingsboard backend (based on Akka actors, not detailed in the figure), transfer the data stream to all the relevant rules. (3) In this scenario, the rule is configured to use the Spark-plugin, so that the stream is transmitted to a Spark Task. (4) As a response, the spark task re-publishes new types of events, such as alarms, or transformed data, in order to be presented in the front-end (5).

Based on the data model described above, and the requirements of our study case (the early detection of *Phytophthora infestans*), the extended architecture described in Fig. 3 was proposed. A first version of the architecture was implemented as a *fork* of the official Thingsboard distribution https://github.com/LIS-ECI/thingsboard, considering the following elements:

– An extension to the default data model (implemented in Cassandra, a NoSQL time-series database) that integrates the concepts of Farm, Land Lot, Crop, and the 'good practices' check list proposed by the International Federation of Organic Agriculture Movements [16].

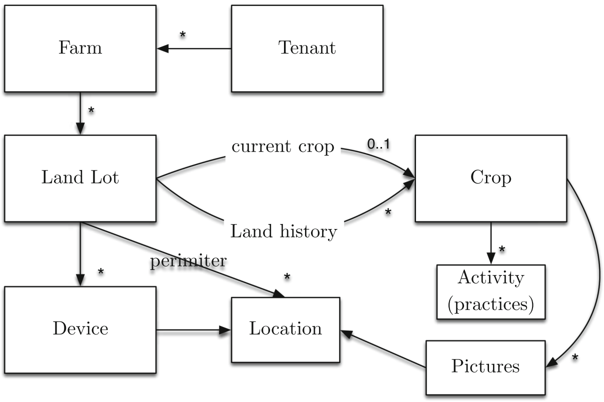


Fig. 2. Extended Thingsboard UI, including new data hierarchy and graphical representation of geo-referenced data.

- An integration of complementary database engines, with a distributed transactions mechanism and an access API: MongoDB for geo-referenced indexing of sensors, crops and farms (Location entity in Fig. 2); MongoDB+GridFS for geo-referenced pictures; and REDIS for keeping temporary-volatile data, such as intermediate states of a rule evaluation.
- An API for the registration of third-party platforms the platform is going to interact with.
- A Framework within Spark for the definition of new rules and actions for a potential phenomenon/disease in the crops monitored by the platform. Such framework allows the definition of rules with access to the extended and complementary data model, and the definition of actions with access to the third-party platforms API.

Extended User Stories. Given the hierarchies of the proposed extended data model, and the guidance of potential Stakeholders, new User Stories[2] and Wireframes[3] were defined, including the registration of sensors and the configuration of dashboards. Figures 4 and 5 show screenshots of two of the user stories developed so far using the Widgets model aforementioned. Figure 4 shows how geo-referenced details are now used to show elements such as the physical distribution of the crops within the farm. Figure 5, on the other hand, shows how the new User Stories allow the farmer to keep tracking of the good practices in a crop over the time [16]. Such information, as mentioned before, could therefore be accessed by the rules registered in Spark.

[2] https://trello.com/b/V6wD9VEX/thingsboard-extensi%C3%B3n.
[3] https://ninjamock.com/s/9W6WWRx.

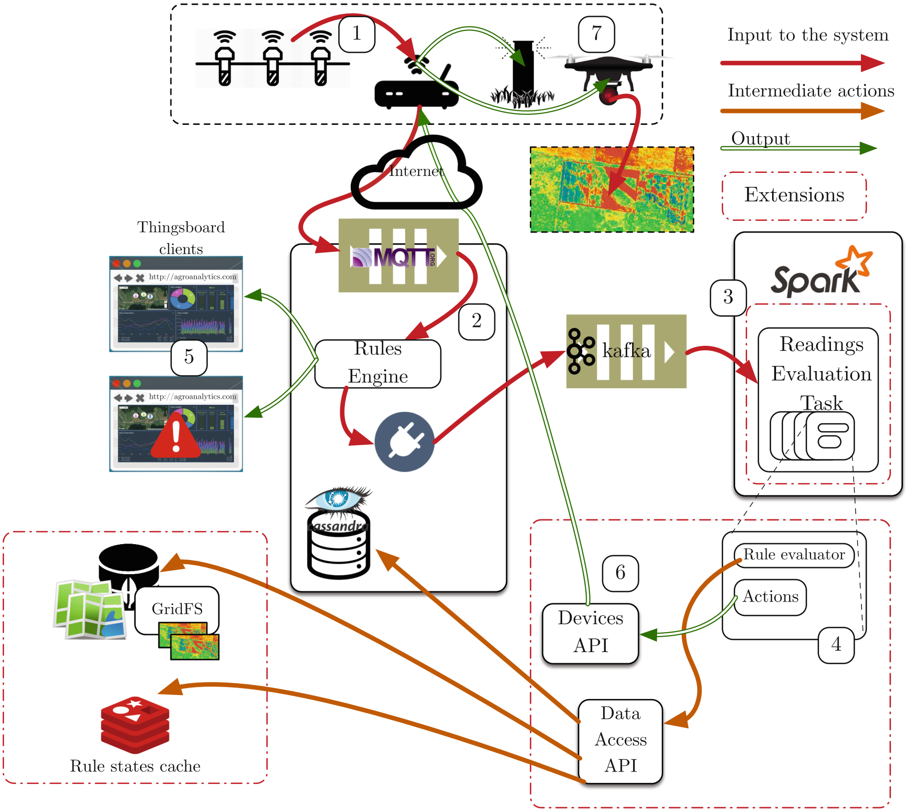


Fig. 3. (1) Sensors transmit data stream (through an Internet-gateway) using one of the protocols currently supported: MQTT, CoAP or HTTP. (2 & 3) Thingsboard backend transfer the data stream to all the relevant rules, in this case, a rule with the Spark-plugin enabled. (3) A Spark Task configured by default to handle all the readings, delegates its evaluation to a series 'Evaluation/Action' components (previously injected to such task). (4) The 'Evaluation/Action' component, based on the sensor readings, crop details, will generate warnings through conventional Thingsboard alarm mechanism. (6) Such 'Evaluation/Action' components would be able to fire actions in third-party systems (e.g. an autonomous drone) providing them with all the details required for their mission. In the figure, the platform fires an autonomous drone (7) that will take multi-spectral pictures of the alarm zone, for further analysis.

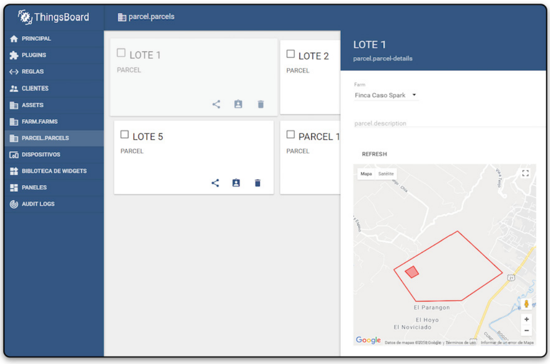


Fig. 4. Extended Thingsboard UI, including new data hierarchy and graphical representation of geo-referenced data.

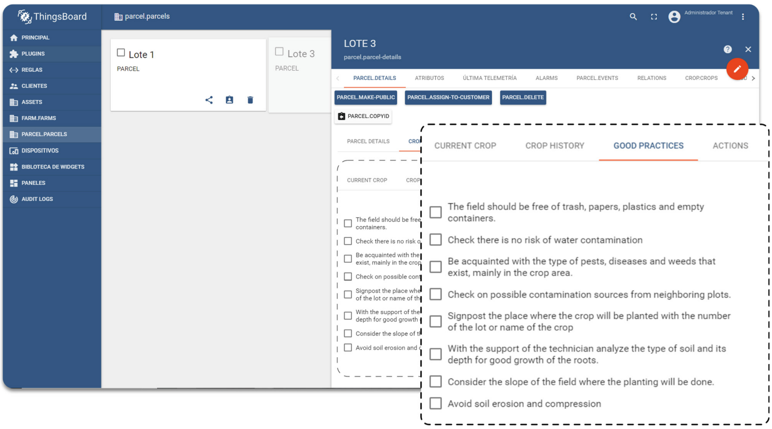


Fig. 5. Extended Thingsboard UI, including new data hierarchy and graphical representation of geo-referenced data.

4 Proof of Concept

This section presents a proof of concept of the architecture extension proposed to the Thingsboard platform in Sect. 3. A simulated scenario for the detection of *Phytophthora infestans* conidia pest was chosen for the experiments. The integration of previous works in control architectures for autonomous robots [6] is expected for future field tests. However, for the testing purposes of this paper (with a software architecture scope), a simulated drone fleet controller (which simply echoes all the received instructions) was integrated as a means to verify the outcome of the proposed scenario. As an outcome of the early detection of the *Phytophthora infestans*, not only the generation of alarms through the platform is expected, but the activation, with the right set of instructions, of the (simulated) drone fleet controller.

4.1 Early Detection of Phytophthora Infestans

There are different models, documented in literature, for the prediction of sporulation of *Phytophthora infestans* conidia pests. This prediction, as a means of early detection, allows applying a timely phytosanitary treatment for the crop in order to mitigate the development of the late blight disease. As the reader could see in this figure, the consequences of the propagation of this disease is catastrophic to the crop.

One of the prediction models is the Smith Period Model [10], where the minimum temperature and relative humidity are data considered. The authors of this model proposed that a *Smith Period* occurs when the minimum temperature is higher than 10 °C and the relative humidity is greater than 90% for 11 h, for 2 consecutive days. When two *Smith Periods* occur it is necessary to perform the first application of a fungicide to mitigate the sporulation risk before the disease appears in the crop. If the temperature and humidity criteria are met only on the first day, and on the second day they reach 10 h of relative humidity greater than 90%, it indicates that only one *Smith Periods* has taken place.

As mentioned before, the proposed extension for the *Thingsboard* platform, makes possible the integration of a model like the former as a software component. For evaluation purposes, the model was implemented as a sliding-window algorithm, depicted in Fig. 6. Such figure, on one hand, shows the importance of the session-persistence feature proposed for the architecture. The details provided in Algorithm 1 show, on the other hand, how the framework within the Spark model enables access to crop's details, including history and geo-referenced information. Furthermore, as shown in Algorithm 2, the framework also enables the definition of actions to be performed when there is an alert confirmation, including the interaction with third-party platforms (in this case, launch a hypothetical drone and notify to tenants of nearby crops).

4.2 Experiment Setup and Results

For our experiments, 25 sensing devices, associated with 5 different crops were simulated through *Gatling* tool, an open source Load and Performance profiler tool. Two of such sensors were fixed to produce data within the range of a series of *Smith Periods*. For testing purposes, time was scaled by a factor of 86.400 to 60 s (1 day=1 min). During the execution of the simulation, the Dashboard of the crop with the fixed sensors started as shown in Fig. 7(a), and few minutes later, generated an alarm (as expected), as shown in Figure 7(b). Moreover, in the same simulation scenario, the alarm is sent in real time to the simulated drone fleet controller with the geographic localization of the field, as shown in Fig. 8. For this setup, the servers where distributed in three virtual machines with 4 GB of RAM, running over an Intel (R) Xeon(R) E5620-2.4 GHz server.

Algorithm 1. Phytophthora-infestans-risk-evaluation

```
procedure RISK-EVALUATION(cropid, humidityData, temperatureData)
Input:
  – cropid: crop's unique identifier.
  – humidityData: average humidity since last reading, provided by a sensor.
  – temperatureData: average temperature since last reading, provided by a
    sensor.
    cropType ← dataapi.getCropType(cropid)
    riskDetected ← False
    conditionsFulfilled ← " − "
    now ← CurrentTime
    if cropType is 'potatoe' and humidityData and temperatureData satisfy the condition then
        conditionsFulfilled ← " + "
    if It's the 1st time receiving data then
        window ← conditionsFulfilled
        cacheapi.saveFirstTime(cropid, now)
    else if Eleven hours have already elapsed then
        window ← cacheapi.getWindow(cropid)
        window.removeFirstElement()
        window+ = conditionsFulfilled
        if the amount of '+' in window is high then
            if It's the 1st day and Not exist a Smith period then
                cacheapi.saveSmithPeriod(cropid, True)
            else if It's the 2nd day then
                riskDetected ← True
                window ← ""
                cacheapi.saveFirstTime(cropid, now)
                cacheapi.saveSmithPeriod(cropid, False)
    else
        window+ = conditionsFulfilled
    if 1st day finished then
        if Not exist a Smith Period then
            cacheapi.saveFirstTime(cropid, now)
        window ← ""
        cacheapi.cacheSmithPeriod(cropid, False)
    if 2nd day finished then
        cacheapi.saveFirstTime(cropid, now)
        window ← ""
        cacheapi.cacheSmithPeriod(cropid, False)
    cacheapi.saveWindow(icrop, window)
return riskDetected
```

Algorithm 2. Phytophthora-infestans-actions

```
procedure PHYTOPHTHORA_INFESTANS_ACTIONS(cropid)
    cropToken ← dataApi.getThingsboardToken(cropid)
    commApi.sendAlertToThingsboard(cropToken)
    cropCoordinates ← geoApi.getParcelCoordinates(cropid)
    neighborsCrops ← geoApi.getCropsInARadiud(cropCoordinates, radius)
    for crop in neighborsCrops do
        apiData.getOwnerData(crop).sendMail("RiskOfPhytophthorainfestans")
    sensorLocation ← geoapi.getSensorLocation(idParcel)
    commApi.sentToThirdParty('droneController', 'applyFungicide', sensorLocation)
```

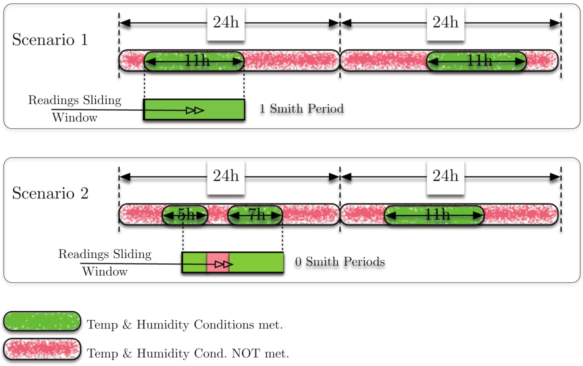
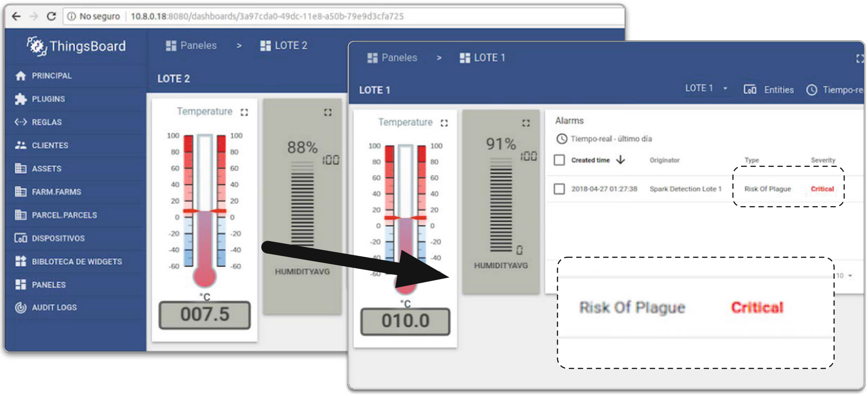


Fig. 6. Sliding Window approach for the evaluation of *Phytophthora infestans* conidia pests based on Smith Periods Model. The platform keeps tracking (in a fixed window) of the positive or negative readings for risk conditions over the last 11 h. When a Window is full of positive readings (Scenario 1), the count of Smith Periods in the 48-hour interval become one. With this approach, long periods of positive readings, with intermediate intervals of negative readings (Scenario 2) could be easily discarded as Smith Periods.

4.3 Performance Evaluation

In order to measure the overhead of the architecture extensions, the same load test were performed over a basic configuration of Thingsboard, with a conventional set of alarms (based on simple value intervals). The outcomes provided by Gatling's dashboard after running the same load test on both configurations, were nearly identical. However, this could be explained given the asynchronous nature of the platform's entry point: an MQTT server. For this reason, all the execution times of the Algorithm 1 were measured over the experiment. As can be seen in Fig. 9, the overhead of the data-access API and third-party systems interaction is in most cases about 0.5 and 1.5 s, with few outlier peaks. Given the low frequency of the data transmitted in most IoT applications, this overhead could be considered negligible.



(a) (b)

Fig. 7. Default dashboard for the simulated environment, including temperature and humidity sensors, before (a) and after (b) the warning generated by the *Phytophthora infestans* evaluation rule.

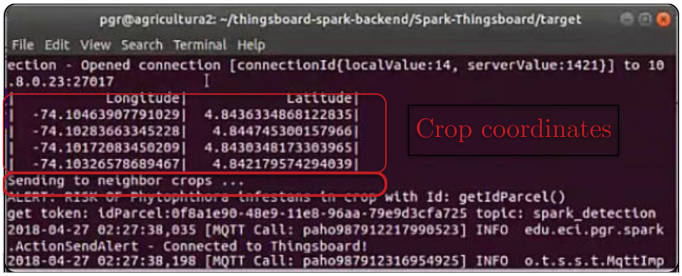


Fig. 8. Screenshot of the simulated Drone-Fleet controller when a message is received after firing the *Phytophthora infestans* alarm.

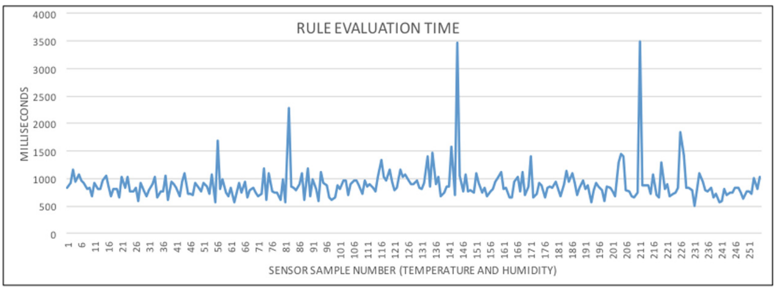


Fig. 9. Differences in milliseconds between the start and end time of each execution of Algorithm 1

5 Concluding Remarks

FAO considers Colombia an important player in the security food policy, but nowadays it does not have enough agricultural technologies in its production processes. In the last five decades, plowable land in Colombia has been disputed by internal war participants and recently by criminal organizations to produce narcotics. Within the framework of the peace agreement signed in 2016, a new perspective for agricultural production is rising for land owners and farmers in terms of crop substitution and the increase of land use as crop fields.

One of the most important tasks for agriculture labors is increasing the uses of modern techniques of production. However, the lack of reliable data about specific varieties of plants in Colombia is a big gap to be filled. With this panorama, the incursion in new technologies like IoT and Analytics is mandatory were the country to want to increase its food exports to the rest of the world.

In this work, the authors have presented an extension of a popular open source platform called *Thingsboard*, that is used to collect and manage data provided by sensors. This extension aims to be the core of a future cloud-based MaaS (Monitoring as a Service) tailored to the needs of the Colombian farming industry. The architecture of the proposed extension has been validated and illustrated with a real-life scenario, where the risk of an extremely dangerous potato disease is identified in real time and a simulated controller of autonomous drones is activated in response. Future work will integrate a module for data exploration (time series, pictures, etc.) and spatial-temporal tagging by experts. In the long term, once enough data has been collected and tagged, new classification models for other diseases would be trained (with such data) and integrated into the platform.

References

1. Ahmed, E., et al.: The role of big data analytics in internet of things. Comput. Netw. **129**, 459–471 (2017)
2. Alvarez Villada, D.M., Estrada Iza, M., Cock, J.H.: Rasta rapid soil and terrain assessment: Guía práctica para la caracterización del suelo y del terreno (2010)
3. Bashir, M.R., Gill, A.Q.: Towards an IoT big data analytics framework: smart buildings systems. In: 2016 IEEE 18th International Conference on IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1325–1332. IEEE (2016)
4. Bonér, J., Klang, V., Kuhn, R., et al.: Akka library. http://akka.io/
5. Bruinsma, J.: World Agriculture: Towards 2015/2030: An FAO Study. Routledge, London (2017)
6. Cadavid, H., Pérez, A., Rocha, C.: Reliable control architecture with PLEXIL and ROS for autonomous wheeled robots. In: Solano, A., Ordoñez, H. (eds.) CCC 2017. CCIS, vol. 735, pp. 611–626. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66562-7_44
7. Espana, V.A.A., Pinilla, A.R.R., Bardos, P., Naidu, R.: Contaminated land in colombia: a critical review of current status and future approach for the management of contaminated sites. Sci. Total Environ. **618**, 199–209 (2018)

8. Fry, W., et al.: Five reasons to consider Phytophthora infestans a reemerging pathogen. Phytopathology **105**(7), 966–981 (2015)
9. Hewitt, C., Bishop, P., Steiger, R.: A universal modular actor formalism for artificial intelligence. In: Proceedings of the 3rd International Joint Conference on Artificial Intelligence, pp. 235–245. Morgan Kaufmann Publishers Inc. (1973)
10. Iglesias, I., Escuredo, O., Seijo, C., Méndez, J.: Phytophthora infestans prediction for a potato crop. Am. J. Potato Res. **87**(1), 32–40 (2010)
11. Jawad, H.M., Nordin, R., Gharghan, S.K., Jawad, A.M., Ismail, M.: Energy-efficient wireless sensor networks for precision agriculture: a review. Sensors **17**(8), 1781 (2017)
12. Poole, J., Rae, B., González, L., Hsu, Y., Rutherford, I.: A world that counts: mobilising the data revolution for sustainable development. Technical report, Independent Expert Advisory Group on a Data Revolution for Sustainable Development, November 2014
13. Lasso, E., Corrales, J.C.: Towards an alert system for coffee diseases and pests in a smart farming approach based on semi-supervised learning and graph similarity. In: Angelov, P., Iglesias, J.A., Corrales, J.C. (eds.) AACC'17 2017. AISC, vol. 687, pp. 111–123. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-70187-5_9
14. Lasso, E., Valencia, O., Corrales, D.C., López, I.D., Figueroa, A., Corrales, J.C.: A cloud-based platform for decision making support in Colombian agriculture: a study case in coffee rust. In: Angelov, P., Iglesias, J.A., Corrales, J.C. (eds.) AACC'17 2017. AISC, vol. 687, pp. 182–196. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-70187-5_14
15. Nuthall, P.: Farm Business Management: Analysis of Farming Systems. Lincoln University, CABI (2011)
16. International Federation of Organic Agriculture Movements (IFOAM): Best Practice Guideline for Agriculture and Value Chains. Sustainable Organic Agriculture Action Network/International Federation of Organic Agriculture Movements (IFOAM) (2013)
17. Peisker, A., Dalai, S.: Data analytics for rural development. Indian J. Sci. Technol. **8**(S4), 50–60 (2015)
18. Sarangi, S., Umadikar, J., Kar, S.: Automation of agriculture support systems using wisekar: case study of a crop-disease advisory service. Comput. Electron. Agric. **122**, 200–210 (2016)
19. ThingsBoard. Thingsboard - open-source IoT platform (2018). https://thingsboard.io
20. Vasisht, D., et al.: Farmbeats: an IoT platform for data-driven agriculture. In: NSDI, pp. 515–529 (2017)
21. Beulens, A.J., Reijers, H.A., van der Vorst, J.G., Verdouw, C.N.: A control model for object virtualization in supply chain management. Comput. Ind. **68**, 116–131 (2015)