

Monitoreo de Riego de Cultivos y Detección de Anomalías con Machine Learning

Facundo Etchanchú, Juan Manuel Bouvier y Tomás Bert
{fetchanchu, jbouvier, tbert}@fi.uba.ar

Departamento de Computación
Facultad de Ingeniería, Universidad de Buenos Aires, Argentina

Abstract. El presente trabajo resume la experiencia de desarrollo de un sistema de detección automática de anomalías en el riego de cultivos. Este proyecto se circunscribe al marco de un Trabajo Práctico Profesional de la carrera de Ingeniería en Informática de la Facultad de Ingeniería de la UBA. Los estudiantes trabajaron con la aprobación de la agtech Ponce Automations[1] para asistir en su misión de brindar un mejor monitoreo del estado del riego de los cultivos de sus clientes. Se describe el diseño, implementación y despliegue productivo de una arquitectura capaz de procesar datos distribuidos en forma de series temporales y su clasificación a través de una red neuronal. La solución fue implantada en producción haciendo uso de una infraestructura IoT. Los dispositivos y sensores se comunican con un satélite y vuelcan los resultados telemétricos en la nube, donde se ejecutan pipelines de procesamiento de flujo masivo de datos para clasificar cada medición y detectar potenciales anomalías.

Keywords: Detección de anomalías · Riego de cultivos · Redes neuronales · Procesamiento masivo de datos · IoT

1 Introducción

Gracias a la rápida proliferación de tecnologías IoT y cloud computing, está emergiendo un nuevo concepto que supone mejoras radicales en las metodologías utilizadas en el sector del agro: el Smart Farming[1]. En particular, una de las principales actividades donde se están focalizando los esfuerzos es el riego de cultivos, cuyos sistemas aún mantienen una presencia humana muy fuerte. En cuanto a esta actividad, el sistema que más adopción tuvo en Argentina en las últimas décadas es el de pivote central, considerado como una de las más grandes innovaciones mecánicas del agro de las últimas décadas[2][3]. Este mecanismo de riego por aspersión está conformado por una pirámide en el centro y una serie de tramos enlazados que giran alrededor de la misma, esparciendo agua y formando un círculo. Si bien los caudales de agua utilizados varían según el cultivo y la configuración del equipo, es común que superen los 150.000 lt/h[4]. Al ser equipos que operan de manera autónoma por varios días, no requieren de personas en el lote. No obstante, se realizan rondas de supervisión diarias ya que cualquier desperfecto en el equipo puede resultar en grandes pérdidas.

2 Authors Suppressed Due to Excessive Length

Ponce Automations es una start-up agtech responsable de programar y distribuir los sensores colocados sobre los mecanismos pivotaes de riego en cultivos[5], con el fin de ofrecer un sistema de monitoreo y supervisión para detectar manualmente fallas. Actualmente se encuentra en estado de reestructuración y revisando su misión; anhelan ser una empresa de servicios personalizados para sus clientes y no sólo proveedores de hardware y software enlatado. Ponce Automations provee una herramienta a los productores que les permite visualizar el estado del riego.

En el presente trabajo, los estudiantes implementaron un servicio escalable y extensible que permite procesar datos enviados por los sensores para evaluar anomalías y emitir alertas al supervisor del riego. A tal fin, y con el consentimiento de Ponce Automations, se analizó la información histórica de riegos y se desarrolló una extensión a los sistemas existentes para procesamiento de mediciones de sensores. Este enfoque permitió un despliegue simple sobre la infraestructura productiva de Ponce Automation en Nube[6] y la posibilidad de realizar un trabajo autónomo, con puntos de contacto con el sistema existente claramente definidos. El sistema permite configurar cualquier canal para enviar la notificación de alertas a los usuarios. Automatizando este flujo y procesamiento de datos se aporta valor agregado tanto al negocio de Ponce Automations como al de los productores.

Para alcanzar dicho objetivo, el documento se estructura de la manera descrita a continuación. En primer lugar, en el **Análisis del problema** se releva el funcionamiento actual del sistema y se identifican las potenciales mejoras a desarrollar. Durante la etapa de **Diseño de arquitectura** se sugiere cómo integrar la solución en forma paralela al funcionamiento productivo. En **Desarrollo** por un lado se describe el análisis y prueba del stack tecnológico a implementar tal que se cumplan los requerimientos previos y, por el otro, se explica el desarrollo del motor de aprendizaje automático para la clasificación de anomalías mediante una red neuronal, basándose sobre algoritmos supervisados tras reiterados intentos fallidos sobre el mundo de los no supervisados. Por último, se describen las **Conclusiones** del proceso entero, destacando la integración productiva del nuevo stack como una solución innovadora, accesible y trascendente para la empresa y alineada a las necesidades y visión de negocio de la empresa.

2 Análisis del problema

Con el fin de identificar los puntos de integración de la solución propuesta, se hizo relevamiento de tecnologías en Ponce, se analizó el ciclo de vida de sus datos, la responsabilidad funcional de cada uno de sus componentes y las tecnologías de cada uno de ellos.

La Figura I describe el camino de los datos, comenzando en los sensores físicos de presión y posición ubicados en el sistema de riego, pasando por los componentes necesarios para persistirlos, hasta finalmente llegar a la interfaz visual para ser consultada por el cliente.

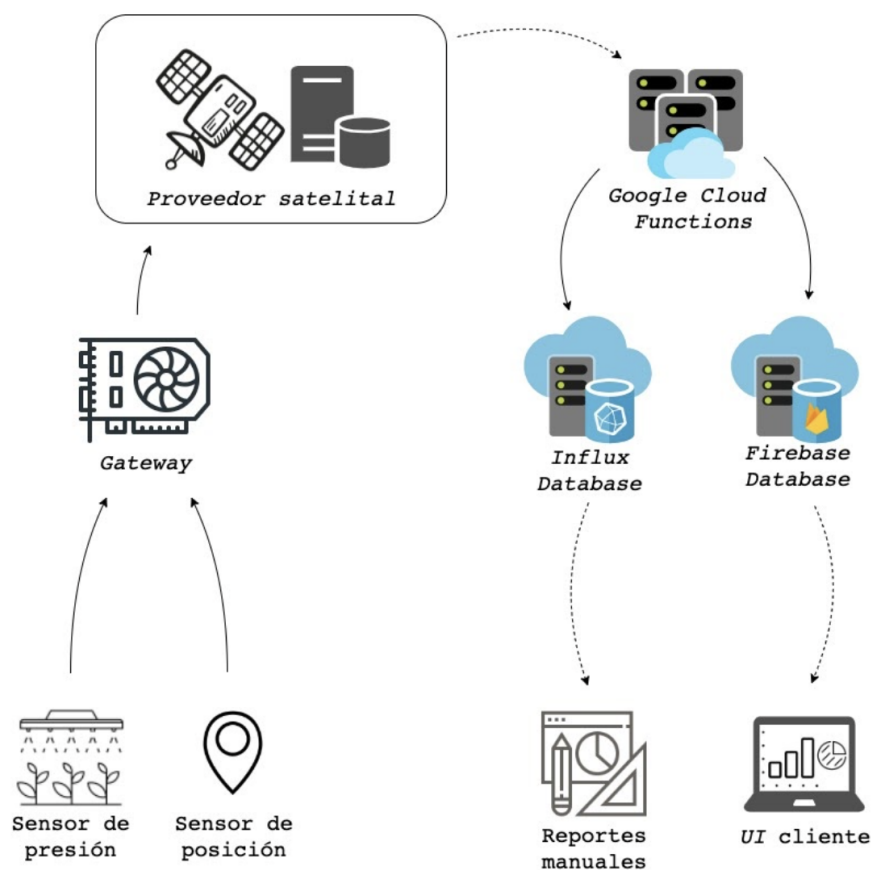


Fig. 1. Flujo de datos del sistema existente. Los datos recolectados por los sensores de campo son exportados al software de Ponce Automation en Google Cloud gracias a un proveedor de soluciones satelitales.

2.1 Objetivos Principales

Luego de una serie de entrevistas con Ponce Automations se definieron los siguientes requerimientos:

- El sistema debe interpretar los datos enviados por los sensores distribuidos en los sistemas de riego y tener la capacidad de aprender automáticamente las anomalías y los comportamientos no esperados de los riegos a través de los datos.
- El cliente debe recibir una alerta en su celular en vivo cuando se detecta una anomalía en el funcionamiento de su pivot.

4 Authors Suppressed Due to Excessive Length

- El sistema permitirá visualizar reportes y gráficos relativos al comportamiento de cada pivot.

3 Diseño de arquitectura

A raíz de los requerimientos y de un conjunto de propuestas, se logró diseñar un sistema compuesto de dos módulos según se observa en la vista lógica de la Fig. 2.

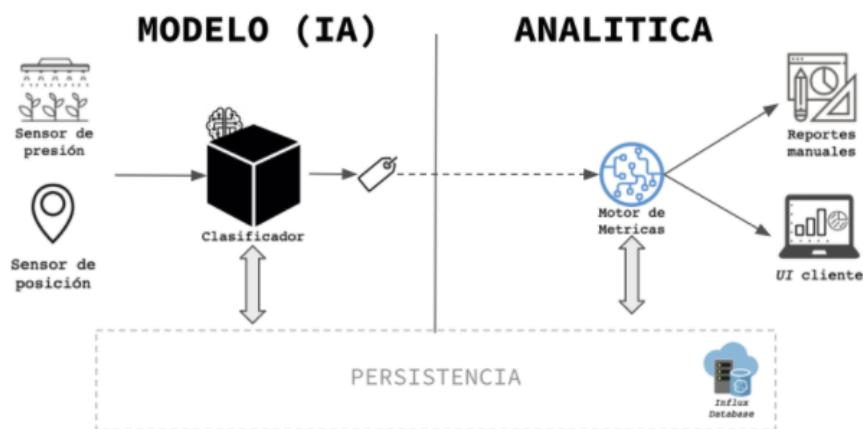


Fig. 2. Prototipo diseñado para el procesamiento y detección de anomalías.

En cuanto al input, la arquitectura propone la ingesta *streaming* [7] de datos (*snapshots*) de forma directa, emitidos por los dispositivos distribuidos productivamente. Luego se clasifican dichos datos, configurando además reglas parametrizables para tener la capacidad de visualizar y alertar sobre comportamientos derivados. Para satisfacer la propuesta, la arquitectura se compone de dos partes principales:

- **MODELO (IA):** Se encarga de clasificar los datos enviados por los dispositivos en tiempo de escritura a la base de datos (*near real-time*). Dentro de esta sección, el componente más importante es el modelo de clasificación automática previamente entrenado y con una alta tasa de acierto a la hora de clasificar. Por lo tanto, el requerimiento fundamental del modelo es categorizar con alta eficacia y eficiencia.
- **ANALITICA:** Proporciona un motor para análisis de los datos. Dicho análisis llega al usuario final a través de una vista la cual muestra distintos

paneles que constan de gráficos y filtros tomados de la base de datos productiva. A raíz de los gráficos y reglas personalizadas se le da al usuario la posibilidad de configurar alertas. Prima construir el componente de forma genérica tal que sea aplicable a distintos usuarios y potencialmente a diferentes negocios sin incurrir en esfuerzos de desarrollo costosos.

A partir del modelo lógico propuesto, se llevó a cabo un análisis de distintas variantes de arquitectura disponibles para su desarrollo. Se investigaron alternativas *SaaS*[8], *PaaS*[9] como *stack*[10] de servicios open-source o, incluso, desarrollo interno sin usar software de terceros.

4 Desarrollo

La implementación se bifurcó en dos servicios, desarrollados en paralelo: el modelo de detección de anomalías *per se* y la infraestructura para soportar el flujo masivo de datos.

4.1 Selección e implementación del *stack* tecnológico

En el marco de proyectos relacionados a dispositivos IoT y procesamiento de datos en la nube, se evaluaron distintas alternativas de stacks tecnológicos. Para tomar la decisión definitiva se puso foco en la minimización de los costos y del impacto en el sistema preexistente de Ponce Automations. Por otra parte, se consideraron también los factores de extensibilidad, costos de desarrollo y costos de mantenimiento. Una opción era implementar todo un stack de cero, brindando una arquitectura totalmente customizable. Otra, aprovechar el uso de GCP por parte de Ponce e implementar la arquitectura propuesta por la plataforma para casos de uso similares. Por último, el uso de la herramienta *open-source T.I.C.K stack* (**TICK**, de aquí en adelante), desarrollada por Influxdata, la empresa creadora de InfluxDB.

Finalmente, se consideró como la mejor opción y aceptada por Ponce Automations el TICK stack. El mismo es un conjunto de servicios que ofrece persistencia, procesamiento de datos en tiempo de escritura y vistas para el monitoreo y configuración de alertas. El nombre del *stack* proviene de los cuatro componentes que lo conforman. En primer lugar, *Telegraf*[11] que es el agente que se encarga de recopilar y reportar métricas y/o eventos desde cualquiera de las fuentes posibles. Por otro lado, *InfluxDB*[12], la base de datos encargada de la persistencia orientada a serie de datos temporales. Luego se encuentra *Chronograf*[13] que permite ver y controlar el estado de todos los servidores del clúster. Ofrece una interfaz gráfica para configurar tableros de métricas, alertas y ejecutar procedimientos (jobs). Por último *Kapacitor*[14] que se trata del motor que facilita la configuración de alertas y ejecución de jobs de flujos de tarea para entradas recientemente escritas en la base de datos.

6 Authors Suppressed Due to Excessive Length

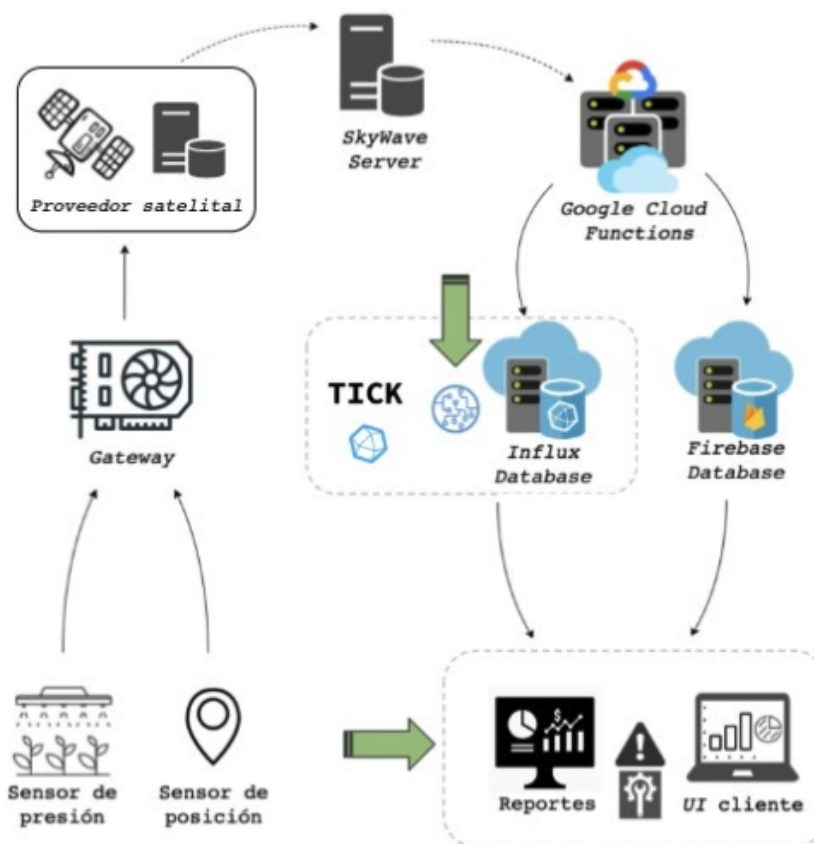


Fig. 3. Integración del *stack* de TICK con el flujo de datos descrito en la Fig. 1. En verde, los puntos de contacto con la arquitectura existente.

La arquitectura implementada aporta gran valor a los procesos de Ponce Automations, permitiendo funcionalidades tales como la manipulación de nuevas bases de datos de series temporales; aportando extensibilidad y escalabilidad a la plataforma para agregar servicios de *machine learning*[15], alertas y reportería; y soportando la integración de cualquier otro mecanismo de procesamiento y transformación de flujos de datos en tiempo de escritura. Todos los servicios desarrollados se centraron en la funcionalidad e integración del sistema de detección de anomalías con la plataforma de Ponce. Para los usuarios que interactúan con la aplicación final, TICK *stack* resulta intangible siendo necesario el diseño, integración e implementación de vistas que permitan que no solo el analista de Ponce, sino también el usuario final, tengan acceso a métricas, reportes y visu-

alizaciones sobre el comportamiento y anomalías de sus dispositivos. Se definió entonces la necesidad de configurar e implementar dos módulos de monitoreo para poder visualizar e interactuar con el sistema de detección de anomalías en riegos:

- **Chronograf:** Mencionado anteriormente, es la plataforma para visualización por parte de Ponce Automation que reemplaza sus reportes internos de poca flexibilidad.
- **Dashboard:** sistema personalizado al cual acceden los clientes de Ponce Automation para visualizar las alertas. Este sistema no cuenta con la versatilidad que muestra Chronograf ya que su fin es poder mostrar los datos que son pertinentes para el cliente final, con el procesamiento correspondiente, y acotado a los dispositivos de su dominio.

4.2 Modelo de detección de anomalías con red neuronal

El dominio del problema se enmarca dentro del universo de procesamiento de series temporales[16] en vivo. En particular, se tiene conocimiento de algoritmos de aprendizaje automatizado no supervisado para el manejo de series de eventos similares[17].

Como primer enfoque entonces se barajaron varios algoritmos de *machine learning* de este tipo y se ejecutaron pruebas de concepto con una batería de alternativas. Los resultados no fueron satisfactorios. El criterio de aceptación más elemental utilizado constó en esta primera instancia de comparaciones entre los reportes de la temporada anterior confeccionados por los ingenieros de Ponce Automations y los resultados arrojados por cada algoritmo. En ninguno de los casos las situaciones detectadas como anómalas manualmente eran identificadas como tales por el motor.

Frente a la necesidad de contar con indicadores para medir el éxito de cada alternativa para ensayar rápidamente cada propuesta y ante la inviabilidad de realizar pruebas productivas con los clientes, se decidió abandonar este modelo. En su lugar se decidió recurrir a algoritmos de aprendizaje supervisado[18] que permitirían contrastar los resultados obtenidos con información conocida como válida

Para ello, fue necesario construir *datasets*[19] en los que ya estuviera catalogado cada punto temporal, indicando si es una anomalía o no. Si bien se podría haber realizado esta clasificación punto a punto consultando los reportes manuales, hubiera resultado una labor muy ineficiente. En cambio, se planteó una métrica nueva para cuantificar qué tan anómalo es cierto punto. La fórmula está inspirada en el indicador estadístico *moving average*[20] tomando como variables independientes la velocidad angular del pivote y la presión, dando como resultado para cada entrada del set de datos un valor continuo entre 0 y 1, donde el 0 significa un caso anómalo y el 1, óptimo. Esta métrica fue constatada contra escenarios reales, elegidos al azar entre el espacio muestral de reportes históricos, y también presentada a Ponce. Sus resultados fueron satisfactorios.

8 Authors Suppressed Due to Excessive Length

$$D_t = \frac{\sum_{i=0}^W w_i * D_{t-i}}{\sum_{i=0}^W w_i} \quad (1)$$

Donde D_t se trata del dato obtenido en el tiempo t , W es el tamaño de la ventana para tomar el *moving average* y w_i es el peso utilizado para el dato D_{t-i} en el ponderado del promedio obtenido con los datos en la ventana.

Una vez obtenido el dataset masivo, fue posible entrenar modelos de machine learning supervisado. Nuevamente, se consultaron casos de éxito de algoritmos supervisados para series temporales y se seleccionaron algunas opciones para llevar a cabo las pruebas de concepto.

Para decidir qué modelo se utilizaría productivamente, tras un exhaustivo análisis se realizó una competencia entre las alternativas más destacables. Se definió como métrica de performance a emplear *F1-score*[21] y se ejecutó cada algoritmo para el dataset productivo, cotejando los resultados con *cross-validation*[22] contra un subset de prueba. El candidato ganador fue la red neuronal conocida como **perceptrón multi-capas**. Finalmente, a este modelo se le ejecutó *Grid-Search*[23] para optimizar sus hiper-parámetros y se obtuvo una eficacia de más de **93%**.

A continuación se presenta la matriz de confusión[24] del modelo construido.

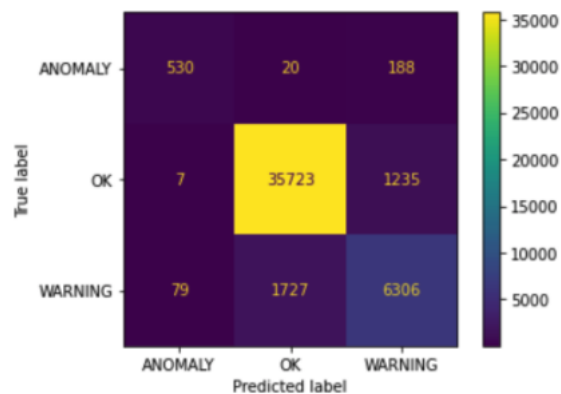


Fig. 4. Matriz de confusión arrojada como resultado de evaluar el perceptrón multi-capas con el *dataset* de prueba empleando *cross-validation*.

Por último, además de medir la eficacia en condiciones estándares, es valioso conocer la flexibilidad de los modelos para responder ante escenarios no ideales, como pueden ser datos faltantes o ruido en la señal. Para este fin, se concibió una serie de tests de robustez para evaluar justamente la tolerancia ante fallos de la red neuronal configurada. En dichas pruebas se midió la eficacia de la red para distintos casos simulados, desde remover aleatoriamente cada vez más datos

hasta inyectar paulatinamente ruido aleatorio a los datos procesados variando tanto rango del ruido como porcentaje de datos afectados. En líneas generales, los scores obtenidos fueron satisfactorios. No obstante, naturalmente la performance ante escenarios con ruido excesivo o gran cantidad de datos faltantes presentó degradación. En síntesis, el modelo resultó lo suficientemente robusto para hacer predicciones fiables ante la presencia de cantidades apreciables -mas no mayoritarias- de ruido o de datos faltantes.

4.3 Monitoreo y Reportes

El sistema incluye un módulo de monitoreo y reportería donde los ingenieros de Ponce Automations pueden evaluar en tiempo cercano al real el estado y comportamiento de los pivotes de riego. Entre otras funcionalidades, se pueden configurar alertas parametrizables que se disparan ante la detección de potenciales anomalías. De esta forma, el producto final automatiza parte del trabajo administrativo y de reportería de Ponce Automations.

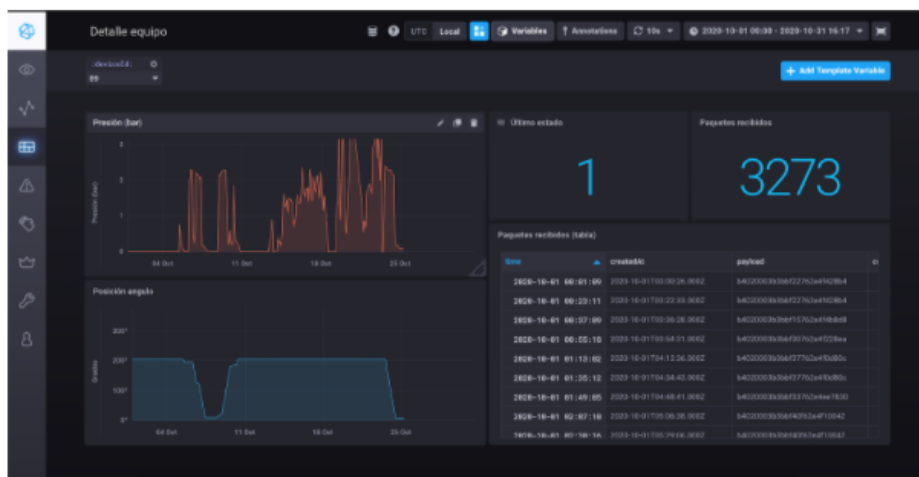


Fig. 5. Interfaz de monitoreo del estado del riego para Ponce Automations, *dashboard* “Detalle equipo” para un dispositivo dado.

Por otro lado, se provee a cada cliente un backoffice desde un sitio web para consultar por su cuenta el comportamiento y estadísticas históricas de sus dispositivos, brindando visibilidad en vivo.

10 Authors Suppressed Due to Excessive Length



Fig. 6. Nuevo *backoffice* del productor para supervisión y reportería de sus dispositivos instalados.

La solución es “plug and play”, lista para ser integrada productivamente y con la documentación técnica relevante y manuales de usuario pertinentes. Además, es extensible y fácilmente iterable, de forma tal que puede adaptarse a las nuevas dinámicas y experimentos que decida encarar la empresa en el futuro inmediato. Cabe destacar que esta solución es un componente integrable de forma paralela sin requerir cambios en el flujo de datos actual de la empresa. Haciendo foco en que Ponce Automations pueda integrar tanto la arquitectura de procesamiento de datos como el modelo de Machine Learning sin necesidad de interrumpir ni modificar su sistema productivo y su entornos de desarrollo, se integró el despliegue de los servicios implementados en el servicio de infraestructura utilizado actualmente: Kubernetes en Google Cloud Platform. Los servicios se despliegan en un cluster de Kubernetes y especificando en su configuración a qué base de datos debe conectarse para tomar registros y procesarlos, visualizarlos o crear alertas puede integrarse al flujo productivo de datos emitidos por los

dispositivos, con sensores distribuidos en los campos de cultivos, de los usuarios de Ponce Automations.

Por último, el desarrollo e implantación de la nueva solución no incurre en costos o recursos extra para la empresa.

5 Conclusiones

Se implementó un sistema de detección automática de anomalías en sistemas de riego de cultivos, plenamente funcional sobre los dispositivos instalados por la empresa Ponce Automations y dentro de su arquitectura preexistente.

La solución es escalable dentro de la nube y extensible en cuanto a que soporta el agregado indefinido de dispositivos recopiladores de métricas. Además, se demostró su flexibilidad y tolerancia a datos incompletos mediante una batería de tests de robustez. Los resultados de las pruebas de concepto fueron considerados exitosos por las partes interesadas, utilizando los datos históricos disponibles hasta la fecha.

Quedó demostrado finalmente que la detección automatizada y reactiva de anomalías en sistemas de riego de cultivos no solo es factible si no que también es viable en zonas rurales con conectividad pobre, en lo cual se envían los datos a la nube a través de tecnologías LoRa[25] proveídas por satélites. A modo de cierre, queda pendiente evaluar la eficacia del modelo propuesto luego de cierto tiempo de uso productivo en todos los dispositivos instalados. Asimismo, es deseable probar para futuras iteraciones nuevos modelos de procesamiento de flujos de datos, ya sea para refinar la solución o para atacar nuevos problemas que requieran manipulación de eventos en tiempos de escritura o por lotes.

Referencias

1. Ponce Automations Página principal, <http://www.ponceautomations.com/>. Accedido el 4 Abr. 2021
2. Dispositivos móviles de riego por aspersión. Wikipedia, 2017. https://es.wikipedia.org/wiki/Dispositivos_moviles_de_riego_por_aspersion. Accedido el 16 Ago. 2019
3. J. M. Tarjuelo Martín-Benito, El riego por aspersión y su tecnología. Ed Mundi Prensa, 2005.
4. Texas Agricultural Extension service. The Texas A&M University system. "Center Pivot Irrigation". <https://aglifesciences.tamu.edu/baen/wp-content/uploads/sites/24/2017/01/B-6096-Center-Pivot-Irrigation.pdf>. Accedido el 23 Ago 2019
5. PIVOTES: La Revolución Circular del Riego, Red Agrícola. <https://www.redagricola.com/cl/pivotes-la-revolucion-circular-del-riego>. Accedido el 10 Nov 2020
6. Computación en la Nube. Wikipedia. https://es.wikipedia.org/wiki/Computacion_en_la_nube. Accedido el 10 Nov 2020
7. Streaming. Wikipedia. <https://es.wikipedia.org/wiki/Streaming>. Accedido el 4 Oct 2020

12 Authors Suppressed Due to Excessive Length

8. Software como servicio. Wikipedia. https://es.wikipedia.org/wiki/Software_como_servicio. Accedido el 4 Oct 2020
9. What is Product-as-a-Service (PaaS)?. Cristina Quijano. <https://www.firmhouse.com/blog/what-is-product-as-a-service-paas>. Accedido el 4 Oct 2020
10. ¿Qué es un stack tecnológico (o tech stack)?. Mixpanel. <https://mixpanel.com/es/topics/what-is-a-technology-stack/>. Accedido el 4 Oct 2020
11. Telegraf Homepage, InfluxData. <https://www.influxdata.com/time-series-platform/telegraf/>. Accedido el 10 Ene 2021
12. InfluxDB Homepage, InfluxData. <https://www.influxdata.com/products/influxdb/>. Accedido el 10 Ene 2021
13. Chronograf Homepage, InfluxData. <https://www.influxdata.com/time-series-platform/chronograf/>. Accedido el 10 Ene 2021
14. Kapacitor Homepage, InfluxData. <https://www.influxdata.com/time-series-platform/kapacitor/>. Accedido el 10 Ene 2021
15. Aprendizaje Automático, Wikipedia. https://es.wikipedia.org/wiki/Aprendizaje_automatiko. Accedido el 20 Oct 2020
16. Series Temporales. Wikipedia. https://es.wikipedia.org/wiki/Serie_temporal. Accedido el 12 Abr 2020
17. Aprendizaje Supervisado, Wikipedia. https://es.wikipedia.org/wiki/Aprendizaje_no_supervisado. Accedido el 15 Abr 2020
18. Aprendizaje No Supervisado. Wikipedia. https://es.wikipedia.org/wiki/Aprendizaje_supervisado. Accedido el 15 Abr 2020
19. Conjunto de Datos (dataset). Wikipedia. https://es.wikipedia.org/wiki/Conjunto_de_datos. Accedido el 15 Abr 2020
20. Media Móvil. Wikipedia. https://es.wikipedia.org/wiki/Media_movil. Accedido el 15 Abr 2020
21. Precision, Recall, F1, Accuracy en clasificación, Jose Martinez Heras <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion>. Accedido el 15 Abr 2020
22. Validación Cruzada. Wikipedia. https://es.wikipedia.org/wiki/Validacion_cruzada. Accedido el 15 Abr 2020
23. What Is Grid Search?, Farhad Malik. Medium. <https://medium.com/fintechexplained/what-is-grid-search-c01fe886ef0a>. Accedido el 15 Abr 2020
24. Matriz de confusión. Wikipedia. https://es.wikipedia.org/wiki/Matriz_de_confusion. Accedido el 20 Abr 2020
25. LoRa. Wikipedia. <https://en.wikipedia.org/wiki/LoRa>. Accedido e 26 Ago 2021