INVESTIGACIÓN e INNOVACIÓN en INGENIERÍAS
ISSN2344-8652

Arquitectura para el análisis de datos agronómicos en un ambiente de Big Data

An Architecture for Assessing Agronomic Data within a Big Data Environment

Luis Felipe Vargas Rojas **Víctor Andrés Bucheli Guerrero**

Universidad del Valle, Colombia

Recibido: 21/05/2020
Aceptado: 23/09/2020
Publicado: 19/11/2020

Correspondencia de autores:
felipe.vargas@correounivalle.edu.co

Resumen

Objetivo: Proponer una arquitectura de Big Data especializada en el proceso de predicción del rendimiento de un cultivo. **Metodología:** Se investigó tomando en cuenta dos enfoques, por un lado, el proceso de predicción de rendimientos de cultivos, por otro lado, se estudiaron arquitecturas de software relacionadas. A partir de las investigaciones se definieron los requerimientos para un sistema de almacenamiento y procesamiento en este ámbito. **Resultados:** la arquitectura incluyó (1) un modelo de datos en colecciones MongoDB; (2) un sistema de encolamiento Kafka; y (3) un sistema de procesamiento en PySpark. La arquitectura hereda de las tecnologías usadas la capacidad de escalamiento vertical y horizontal, de atender datos heterogéneos y variables de dominio específico, además de permitir la interacción con diferentes transformaciones y modelos de aprendizaje automático. **Conclusión**: Las tecnologías de Big Data pueden modelar el proceso de predicción de rendimientos de cultivo, este esquema sirve como referencia para llevar a cabo análisis de datos agronómicos sobre un ambiente de Big Data escalable y flexible.

Palabras clave: Análisis de datos, Datos Agronómicos, Big Data, Apache Spark, MongoDB

Abstract

Objective: This work seeks to propose a big data architecture targeted toward crop yield prediction processes. **Methodology**: A two-fold research methodology was used herein. On the one hand, it focused on crop yield prediction processes, and on the other hand, several related software architectures were assessed. Accordingly, the implementation requirements for a storage and processing system were defined within the given context. **Results:** The resulting architecture included (1) a MongoDB collections data model; (2) a Kafka queueing system; and (3) a PySpark processing system. From these technologies, the developed architecture inherits the capacity for vertical and horizontal scaling, to serve heterogeneous data and domain-specific variables, as well as to facilitate interaction with different transformations and machine learning models. **Conclusions:** Big data technologies can be used to properly model crop yield prediction processes. This scheme serves as a reference for conducting agronomic data analysis on a scalable and flexible big data environment.

Keywords: Data Analysis, Agronomic Data, Big Data, Apache Spark, MongoDB

Como citar (IEEE): L. Vargas-Rojas., y V. Bucheli-Guerrero. "Arquitectura para el análisis de datos agronómicos en un ambiente de Big Data", *Investigación e Innovación en Ingenierías*, vol. 8, n°2, 233-250, 2020. DOI: https://doi.org/10.17081/invinno.8.2.4211

Introducción

En el 2009 la organización para la alimentación y la agricultura de los Estados Unidos reportó que la producción agrícola debe incrementar un 70% para sostener una población que se espera exceda los 9 mil millones en el 2050 [1]. Esta alerta sumada a los problemas causados por el cambio climático compromete la seguridad alimentaria. Para mitigar el impacto causado, los procesos agrícolas se deben apoyar en sistemas de análisis de datos más exhaustivos. Uno de los problemas de interés en este dominio es la predicción del rendimiento de un cultivo (PRC), el rendimiento de un cultivo es una cantidad obtenida en la cosecha que se mide en kilogramos por hectárea. El PRC se ve afectado por la interacción GxE (Gen & Ambiente), estas dinámicas son difíciles de modelar por su complejidad y comportamiento no lineal, además de que dependen de un gran número de dimensiones a menudo desconocidas y de tipos cuantitativas o cualitativas [2]. Entendiendo este problema estadísticamente como un problema de regresión algunos modelos de aprendizaje estadístico y automático se han desarrollado, por ejemplo: modelos de redes neuronales [2], árboles de decisión [3] y regresión lineal [4].

La integración de experimentos con diferentes formatos y la capacidad de recolectar información más detallada de la temporada de crecimiento del cultivo convierten este dominio en un espacio propicio para aplicar Big Data [5]. Por ejemplo, el volumen, por los datos de sensores; la variedad, en efecto a datos de sensores, hojas de cálculo, sistemas de información, bases de datos tradicionales; la velocidad, para procesar datos a tiempo y evitar plagas o pérdidas por condiciones climáticas drástica.

Este documento se organiza de la siguiente manera: en la sección estado del arte se muestran los trabajos relacionados, separando entre los orientados a análisis de modelos de predicción, y las propuestas de desarrollos de software o arquitecturas. En la sección metodología se definen los requerimientos, los componentes y las tecnologías seleccionadas para la arquitectura. En la sección resultados se expone detalladamente la arquitectura. Finalmente, en la sección conclusiones se presentan algunos análisis y discusiones.

Estado del arte

En esta sección se estudian dos enfoques de sistemas de predicción. Por un lado, investigaciones donde el interés es comparar los errores de predicción, por otro lado, estudios donde se implementan componentes o plataformas de software.

Modelos de regresión para rendimientos de cultivos

Estos modelos tienen un fuerte componente estadístico y en la calidad predictiva, generalmente se integran los datos en un archivo CSV que se convierte en la base de datos del modelo. Dada la gran cantidad de variables que inciden en esta predicción se puede decir que es un problema de regresión multidimensional [4], la función a alcanzar se puede definir como:

$$F(G,E,M) \quad (1)$$

Donde G agrupa las variables referentes a las características genéticas de un cultivo, E se refiere a variables establecidas por el ambiente donde se realiza la siembra (clima, suelo), y M se refiere a las prácticas de manejo que se establecen antes y durante el crecimiento del cultivo. Esta función retorna un valor continúo denominado rendimiento de un cultivo, algunas veces llamado producción de un cultivo (crop yield).

Tabla 1. Algoritmos de aprendizaje utilizados para la predicción del rendimiento de un cultivo. +

Modelo	Algoritmo	Referencia
Regresión Lineal	Múltiple	[3] [6]
	Robusta	[7]
Kernel	SVR	[3] [4]
Arboles de regresión	CART	[4]
	M5-Prime	[3]
Redes Neuronales	MLP	[7] [2] [8] [4] [9]
	RBF	[4]
Ensamble	RF	[10] [1]
	CIF	[10] [11]

-

Fuente: Elaboración propia

En la Tabla 1 se resumen los algoritmos utilizados, cada algoritmo se basa en un modelo de representación que afecta directamente en el tipo de problema que puede tratar, estos ofrecen ventajas y desventajas. Se diferencia en esta tabla entre modelos estadísticos y modelos de aprendizaje automático. Cualidades como la capacidad de atender datos atípicos, o la facilidad de interpretación, son algunos de los criterios de comparación en cada estudio. En [6] se muestra un cuadro comparativo detallado de los algoritmos.

Regresión Lineal

A pesar de no ser propiamente un algoritmo de aprendizaje automático la regresión lineal múltiple (MLR) ha sido utilizada para la PRC [3], MLR es una técnica estadística la cual se aplica para predicción de una variable dependiente Yi, usando un conjunto de variables independientes explicativas Xij en [7] se define como:

$$Y_i = \sum_{j=1}^{K} B_{i,j} + \varepsilon_i \qquad (2)$$

Donde k representa la cantidad de variables explicativas, Bj es el coeficiente de regresión j y Xij es el valor j de la observación i.

En [3] se aplicó este modelo a un conjunto de datos de distintos cultivos, el conjunto más grande era de 2617 ejemplos. El rendimiento promedio que se obtuvo en esta investigación para este algoritmo fue de RMSE (%) 5.41, y RRSE (%) de 86.18. Se utilizó WEKA como herramienta de análisis y no se encuentra información del tiempo de entrenamiento.

En [8] se utilizó otra técnica denominada regresión lineal robusta, un caso de la regresión lineal múltiple que explota la información tanto como sea posible sin remover los datos atípicos, este método es apropiado cuando los datos tienen un Leverage (que tan lejos está una variable de su media) alto, y se encuentran datos atípicos, esta técnica es importante cuando se violan las suposiciones de la regresión de mínimos cuadrados. La métrica R2 para este modelo fue de 0.65 explicando más del 60% de la variabilidad. Para este análisis se contó con 254 registros, no hay información de la herramienta utilizada ni de los tiempos de entrenamiento.

Redes Neuronales

Por lo general es el método favorito para la PRC especialmente la red de tipo perceptrón multicapa (MLP) con propagación hacia atrás. A diferencia de la regresión lineal este método puede generar un mapeo entre las entradas y las salidas que no es de tipo lineal. Para enfrentar el problema de sobreentrenamiento, que se relaciona con un rendimiento perfecto sobre el conjunto de entrenamiento, pero un rendimiento pobre en datos de prueba, en la literatura encontrada se ha utilizado la validación cruzada [9].

En [10] se utilizó como medida de rendimiento de la red la media de los errores cuadrados. En este trabajo se compararon diferentes topologías de red variando la cantidad de neuronas por capa de 2 a 32 neuronas, un resultado importante obtenido es que se carece de una evidencia clara para afirmar que con redes de tamaños más grandes se obtienen mejores predicciones. En esta investigación se contó con 5241 registros, en el documento no se habla acerca de la herramienta usada para el análisis, ni de los tiempos de entrenamiento.

Arboles de regresión

El aprendizaje de árboles de decisión es un paradigma de aprendizaje inductivo: un modelo se construye desde los datos o partir de observaciones de acuerdo a algunos criterios. El modelo tiene como objetivo aprender una regla general de las instancias observadas [4]

MLR genera modelos globales; sólo hay una fórmula que se adapta al espacio de ejemplos. Mientras que los árboles de regresión utilizan una aproximación diferente, dividiendo recursivamente el espacio de ejemplos, hasta llegar a regiones lo suficientemente pequeñas para ser representadas por un simple modelo [11]. El primer nodo en el árbol (root node) no tiene aristas entrantes, los demás tiene exactamente una arista entrante [3]. Un nodo con aristas salientes se denomina nodo de prueba, y un nodo sin aristas salientes se denomina nodo hoja. Cada nodo hace una división del conjunto de datos en dos o más subespacios, aplicando como criterio de división medidas de impureza (desviación estándar, o el índice Gini). El nodo hoja asigna un valor numérico a la última partición de ejemplos. La elección de los nodos de prueba se basa en la respuesta a la pregunta ¿cuál atributo debe ser examinado en el tope del árbol?, para responder esta pregunta cada atributo es evaluado para definir qué tan apropiado es para separar los datos [4]. El proceso de predecir un nuevo valor se basa en la navegación en este árbol desde el nodo raíz hasta llegar a un nodo hoja. Los algoritmos utilizados varían en tres aspectos: 1) la medida de impureza en atributos continuos, 2) la regla de reducción, y 3) el mecanismo para determinar el valor del nodo hoja, por ejemplo, en M5 es utilizada la reducción de desviación estándar como criterio de impureza, mientras que en CART es usada la varianza.

En [3] se utilizó este método en su variación M5-Prime, se examinaron para varios conjuntos de prueba con un máximo de 2617 ejemplos. Se obtuvieron los siguientes resultados en promedio: RMSE de 5.14, y RRSE (%) de 79.46. La herramienta de análisis utilizada fue WEKA. No hay información de los tiempos de entrenamiento.

En [4] se utilizó Matlab la implementación de classregtree, los datos provenían de varios conjuntos de entrenamiento cada uno representaba un área de siembra específica, el más grande tenía 5241 registros. Se obtuvo de este conjunto de entrenamiento un MAE de 0.4 y un RMSE de 0.9 en el mejor de los casos. No hay registro sobre los tiempos de entrenamiento.

Vectores de Soporte Regresión (SVR)

Máquinas de soporte vectorial (SVM) es un método de aprendizaje supervisado descubierto por [12]. Para la tarea de regresión hay una variante llamada SVR [13]. Dado un conjunto de entrenamiento, la meta de SVR es aproximar una función lineal. Esta función busca minimizar una función de riesgo empírico definido como:

$$R_{emp} = \frac{1}{2}\sum_{i=1}^{N} L_{\varepsilon}(y - f(x)) \quad (3)$$

Donde $L_{\varepsilon}(y - f(x)) = \max((|\xi| - \varepsilon), 0). |\xi|$. es la llamada variable de holgura (slack). la cual se introduce principalmente para tratar de otra manera restricciones imposibles en los problemas de optimización. Usando esta variable los errores son básicamente ignorados mientras que sean tan pequeños como un error propiamente seleccionado . Esta función recibe el nombre de ε-insensitive loss function.

En [3] utilizan SVR obteniendo un promedio de RMSE (%) 5.02, y un RRSE (%) de 81.97. No se precisa en los tiempos de entrenamiento.

En [4] SVR demuestra un buen rendimiento con un MAE de 0.32, y un RMSE de 0.47.

Random Forest

Random Forest (RF) [14] es un algoritmo de aprendizaje de tipo ensamble que puede ser usado para clasificación o para regresión. RF fija un ensamble de modelos de árboles de decisión a un conjunto de datos. El valor que predice para un problema de regresión se basa en el cálculo de la media de los resultados generados por cada árbol. RF puede ofrecer un ranqueo de la importancia relativa de cada variable predictiva. La importancia de las variables se basa en el error de predicción de la regresión, del out-of-bag también llamado OOB.

En [1] utilizan RF para medir el rendimiento de la caña de azúcar, específicamente el paquete de R llamado randomForest, se utilizaron 500 árboles que se derivaron de 500 conjuntos de datos Bootstrapped. Los puntos de separación se calcularon de un subconjunto aleatorio de todas las variables predictivas disponibles. Para este estudio casi el 30% de los datos fueron OOB y no fueron usados en la construcción del árbol. El error de predicción es calculado usando MSE, este modelo logró explicar alrededor del 70% de la variabilidad de los datos. No hay un valor exacto de la cantidad de datos solo se dice que el modelo se hizo usando datos de 1992 hasta 2013.

Arquitecturas y sistemas

Este tipo de publicación tiene un interés en el modelamiento y la estructura de los datos, algunas investigaciones explican los detalles en el uso de redes de sensores inalámbricas y sensores remotos, se proponen arquitecturas y generalmente se apoyan en los sistemas de bases de datos relacionales.

En [15] se enfocan en el despliegue de una arquitectura de red de sensores para un sistema de monitoreo del campo donde crece el cultivo, se desarrolla un sistema de recolección de datos agronómicos en tiempo real de bajo costo, confiable, y con una infraestructura simple, el artículo se enfoca en la parte del hardware y la configuración de la red para transmisión de datos, así como en el consumo de batería para garantizar autonomía prolongada. Este experimento se realiza en una empresa de horticultura ecológica en la ciudad de Murcia, España. Las variables recolectadas fueron: temperatura, humedad, conductividad, y salinidad del suelo, mientras que del aire se midió la humedad relativa, y la temperatura. Finalmente, se creó un pozo a una distancia cercana del campo para medir la conductividad eléctrica del agua y la temperatura. Los datos generados son almacenados en una base de datos MySQL, se crea una aplicación para monitoreo compuesta por: (1) GUI donde se pueden observar los nodos ubicados en un mapa, y los datos recolectados por cada sensor (2) un módulo recolector de datos que espera por el envío, en una conexión dedicada que envía un evento indicando la actualización de los datos en un puerto serial (3) una base de datos con información detallada de los nodos, sus posibles agregaciones, los sensores integrados a cada nodo, los registros históricos, los tipos de sensores y las alarmas históricas recibidas (por ejemplo la falla de batería). Se muestra en los resultados el detalle de los consumos en red y energía, los picos más altos se generan en la adquisición de datos, pero son de corta duración. Antes las mediciones se hacían de manera manual, cada mes una persona con dispositivos portables media variables agronómicas a una frecuencia baja y en la que se pierde información descriptiva de lo ocurrido, luego de la implementación del sistema se puede conocer en tiempo real el estado del cultivo verificando que se cumplan las condiciones agronómicas óptimas para el crecimiento de la planta y se puede estar alerta para tomar medidas ante situaciones adversas

En [16] se propone la creación de una bodega de datos, teniendo en cuenta las características específicas de los datos para la agricultura, su heterogeneidad y distintos niveles de detalle, el diseño propuesto se basa en el modelo en estrella de dimensiones y hechos propuesto por Kimball, las fuentes de datos fueron endógenas provenientes de operaciones internas de la empresa y exógenos provenientes de datos del gobierno y estaciones de clima, un total de 29 conjuntos de datos cada uno con 18 tablas, y con un promedio de 1.4GB en tamaño , los datos se recogieron de distintos países de la Unión Europea y corresponden a cientos de miles ensayos de campo, el diseño incluye hechos referentes a rendimiento, tratamientos, operaciones, y comercio, además de la asociación a múltiples dimensiones como suelo, estación de clima, cultivo entre otros, múltiples hechos pueden tener dimensiones comunes que se utilizan para relacionarlos, el artículo no muestra resultados ni implementaciones.

En [17] se propone un Framework para la predicción de rendimiento de cultivos, se define un paso a paso para la predicción, en el paso 1 se selecciona un cultivo, en el paso 2 se seleccionan las variables independientes, en el paso 3 la dependiente en el paso 4 se selecciona el conjunto de datos y en el paso 5 se hace un preprocesamiento de la variable de salida para discretizar de tal manera que el rendimiento se mide como: alto, regular, y bajo. En este artículo se combinan datos de sensores remotos resumidos a índices como, NDVI, VCI, TCI con datos relacionados a clima y suelo, los algoritmos usados fueron OBIA para sensores remotos, J48 y árboles de decisiones para aplicar minería a las condiciones agrícolas. Las variables recolectadas fueron: temperatura, radiación solar, humedad, lluvias, tipo de suelo, variedad de la

semilla, tipo de fertilizante, cantidad, manejo de maleza, manejo de plagas, longitud del terreno, patrón de siembra y espacio entre plantas, no explica resultados en términos de la ejecución de los modelos.

En [18] se propone un sistema para medir el impacto de la agricultura intensiva al medio ambiente, se analizan datos geoespaciales con analíticas de Big Data. Este estudio se realizó en Cataluña, España por ser una región con altos índices de contaminación a causa de la excesiva ganadería. Se describe una plataforma online llamada AgriBigCAT que usa información de múltiples fuentes combinando análisis geoespacial con tecnologías web, para el almacenamiento de datos se usó Apache Hive, para el análisis de los datos se usó ArcGIS con Hadoop y finalmente la visualización se desarrolló en la plataforma implementada, los resultados de la aplicación y los análisis se estudian a partir de mapas, la interactividad consiste en que un usuario selecciona una región, luego un animal, y finalmente un contaminante, de esta manera se puede observar cómo en regiones de mayor producción ganadera y de déficit de tierra cultivable se presenta mayor concentraciones de contaminantes nitrogenados.

En [19] AgDataBox-API, dado el problema de organización y administración de los datos se propone esta API de acceso HTTP, esta aplicación busca centralizar los datos, y minimizar la complejidad que viene en este campo por el manejo de distintos formatos, desde datos georreferenciados, hasta sensores de imágenes remotos, mencionando una arquitectura modular que permite la fácil integración de distintas aplicaciones externas, se realizó con tecnologías de libre acceso y con Postgres como base de datos.

Metodología

Requerimientos

Una vez reconocidos los trabajos relacionados en el tema de predicción de rendimientos que comprometen datos ambientales y fenotípicos, al hacer la revisión del proceso, tipos de datos y las arquitecturas propuestas se evidencian los retos derivados para la construcción de un sistema de almacenamiento. Se señalan algunos aspectos generales, las variables utilizadas no son fáciles de categorizar, en cada experimento se observaron predictores distintos; mientras que en las arquitecturas no se observaron implementaciones de Big Data con código abierto y no hay un marco de referencia claro para los macro procesos; los experimentos son aislados los datos y los resultados no son replicables fácilmente.

- Para un almacén de datos se debe considerar lo siguiente [16]:
- Debe ser flexible para almacenar datos heterogéneos.
- Debe permitir conectar diferentes fuentes de datos.
- Debe permitir escalabilidad horizontal y vertical.
- Debe facilitar la conexión de herramientas de procesamiento de datos.

El Big Data definido por las 5V [20], velocidad, veracidad, volumen, variedad, valor, en los últimos años muestra una comunidad muy activa trabajando en propuestas de sistemas de almacenamiento, es de interés en este dominio los avances relacionados con estos sistemas en IoT (Internet of Things), donde se observan propuestas de sistemas desacoplados que son accedidos por sistemas de procesamiento ya sea para correr modelos de predicción o incluso para generar Dashboards, visualizaciones, y análisis descriptivos [21].

Para un sistema de procesamiento se deben tener en cuenta los siguientes elementos:

- Debe ofrecer la posibilidad de ejecutar modelos de aprendizaje automático tipo regresión.
- Debe permitir la creación de Pipelines para preprocesar y armonizar los datos
- Debe permitir al usuario interactuar con diferentes escenarios para entender y explorar los datos.
- Debe garantizar escalabilidad para las tareas de procesamiento y modelización

Tabla 2. Tipos de datos involucrados en las temporadas de crecimientos de los cultivos.

Categoría	Frecuencia	Tipo	Descripción
Observaciones	Hora/Dia/Minutos	Numérico	Datos de sensores o estaciones de clima
Objetos	Hora/Dia	Binario	Imágenes de sensores remotos
Contexto	Temporada completa	Textos, Categórico	Tipo de clima, ciudad, área definida, localización de la finca, tipos de suelos, altura
Manejo	Temporada completa/Mes/Día	Textos, Categórico	Fertilizantes, distancias entre siembras, frecuencia de riegos

Fuente: Elaboración propia

En la Tabla 2 se muestran semánticamente los tipos de datos que inciden en la temporada de crecimiento de los cultivos, se distingue de ellos la frecuencia, el tipo de dato y la entidad que representan. Los datos pertenecientes a observaciones tienen un nivel de detalle mayor mientras los datos descriptivos tienen una velocidad de generación menor y representan un solo registro para toda la temporada de crecimiento.

Tabla 3. Entidad base para el ensayo de cultivo.

ID	Lugar	Manejo	Cultivar	Fecha de siembra	Fecha de cosecha
	¿Dónde?	¿Cómo?	¿Qué?	¿Cuándo?	¿Cuándo?
	Rendimiento ¿Cuánto?				

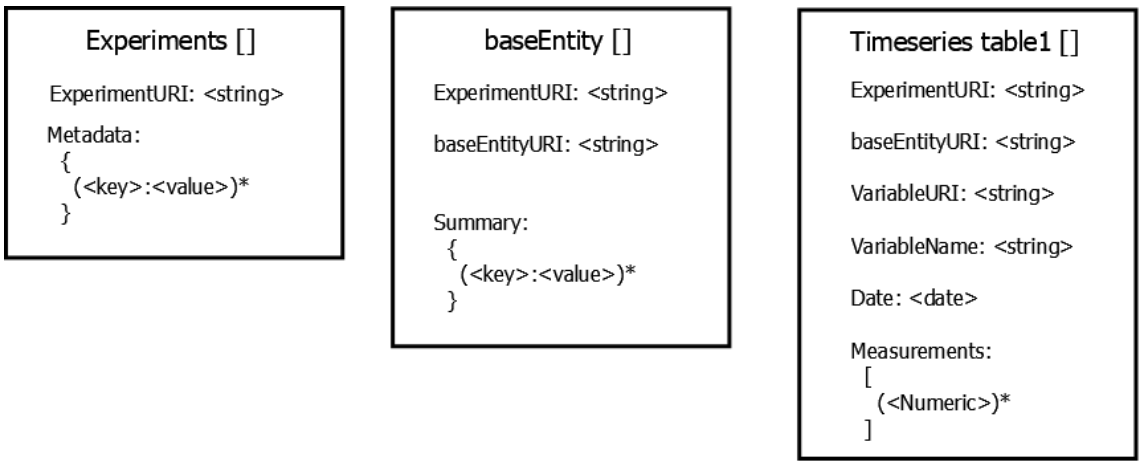
Fuente: Elaboración propia

Todos estos datos se agrupan a partir de una entidad base ver Tabla 3, la entidad base describe el evento, un evento de cultivo se entiende como todo lo que ocurre entre la siembra y la cosecha de la planta, asociado con un rendimiento obtenido [2].

Modelo de datos

El modelo de datos toma en cuenta lo discutido en la Tabla 2 y se apoya de los términos utilizados en el estándar para experimentos de agricultura AgMIP [22].

Figura 1. Modelo de datos para colecciones MongoDB



Fuente: Elaboración propia

En la Figura 1 se observa el modelo de datos para entidades con frecuencia baja, media, y alta, la cantidad de experimentos en un semestre no supera las decenas, por eso se consideran de frecuencia baja, un experimento tiene metadatos asociada al contexto donde se evalúa, las empresas involucradas, los objetivos, fechas de inicio y fin, entre otros [23]. El siguiente nivel, la colección entidad base, puede referirse a una planta, o a un área plantada, este segundo nivel de identificación permite agrupar los datos que en términos de modelos de aprendizaje se convertirán en los registros para entrenamiento de los modelos, se puede considerar esta colección un evento de agricultura ver Tabla 3. El tercer nivel se refiere a variables de series de tiempo, estas son variables que se miden a diario o a veces por horas, para efectos de este sistema el nivel de detalle es el día, se sugiere definir una tabla para cada información sobre series de tiempo, por ejemplo, una para el clima, otra para datos de análisis de imágenes, y otra para eventos de manejo. La flexibilidad del esquema radica en la capacidad de agregar objetos heterogéneos, al mismo tiempo que se proponen tres colecciones como base para armonizar los datos.

Microservicios y ETLs

Un ETL que viene de las siglas en inglés de extractions transformations and load, en el contexto de esta arquitectura se encarga de establecer la conexión con una entidad externa a través de socket, HTTP, o cierto protocolo. Es independiente del lenguaje de programación y cuenta con su propia base de datos, por eso se considera un microservicio. Recibe los datos de la entidad externa los organiza y les aplica los formatos necesarios para finalmente depositarlo en el sistema de encolamiento. El sistema de encolamiento usa

el modelo publish-subscribe, todo microservicio ETL conectado debe tener un cliente que se denomina productor para conectar con el sistema de encolamiento y solamente puede realizar el llamado publish. Se sugiere en la arquitectura el modelo de microservicios porque de esta manera el desarrollador puede construir el proceso en el lenguaje de programación de mayor agrado y se conserva un sistema desacoplado e independiente, se hace más fácil el debug de errores y la mantenibilidad del mismo [24], se puede tener equipos diferentes de desarrollo trabajando en diferentes ETLs.

Sistema de encolamiento

Se elige un sistema de encolamiento mediador a la transferencia de datos para evitar pérdida de datos y para manejar la frecuencia de inserciones dentro del sistema de almacenamiento, la complejidad del sistema de encolamiento puede ser tan sencilla como un Buffer de memoria hasta bases de datos o sistema de archivos como almacenes temporales. Este sistema protege el sistema de almacenamiento dado que es el único medio autorizado para publicar datos en el almacén. Diversos controladores han sido creados en los principales lenguajes de programación para la implementación de clientes (productores o consumidores), de tal manera que el Microservicio ETL solo debe incorporar dicho controlador para enviar los datos, haciendo el sistema interoperable y de fácil comunicación.

Almacenamiento en el clúster

El sistema almacén es un componente donde se reciben todos los datos listos para la integración en colecciones, la complejidad depende de la implementación. Se propone para este punto usar una base de datos noSQL y sin exigencias a priori de esquemas para almacenar los datos. Este microservicio crea un cliente consumidor del sistema de encolamiento es el único componente autorizado para extraer datos del encolamiento.

Base de datos MongoDB

MongoDB se alinea mejor que otras tecnologías de Big Data al contexto de la agricultura, en la Tabla 4 se muestran los criterios de comparación [25] [26].

Tabla 4. Criterios de comparación entre MongoDB y otras tecnologías de Big Data.

Criterio	MongoDB	Cassandra	HDFS
Teorema CAP	CP	AP	AP
Modelo	Maestro esclavo	Peer to peer	Maestro esclavo
Framework de agregaciones	(+)	(-)	(-)
Funciones Map-Reduce	(+)	(-)	(+)
Consulta de datos específicos	(+) soporta dos niveles de indexación	(+-) Solamente con el índice de la fila	(-) requiere lectura de todos los datos
Lenguaje de consulta	(-) lenguaje complejo	(+-) Similar a SQL pero sin JOIN	(-) Busqueda en ficheros
Flexibilidad y esquemas complejos	(+) JSON	(-)	(-)
Soporte para IoT	(+)	(+)	(-)
Escalamiento	(+)	(+)	(++) hasta petabytes

Fuente: Elaboración propia

(+) Cumple, (-) No cumple, (+-) Parcialmente

Dado que el interés concreto son los datos de la agricultura, algunos criterios de la Tabla 4 son vitales para este tipo de datos. Flexibilidad de esquemas y datos complejos, los datos son heterogéneos, el esquema utilizado en AgMIP [22] se basa en JSON, estas complejidades se pueden manejar muy bien con el diseño de colecciones de MongoDB respondiendo al requerimiento R1. Consulta de datos específicos, los datos en la agricultura requieren ser leídos de manera interactiva, y en ciertas unidades de tiempo, no siempre se necesita una lectura completa de los datos sino un análisis enfocado en cierto momento o dimensión, de esta manera poder complementar el requerimiento R7 del procesamiento. Framework de agregaciones, los datos de la agricultura vienen con formatos muy distintos, contar con un Framework para preprocesamiento dentro de la base de datos facilita los pasos previos al análisis de datos. Por otro lado, el tiempo real y la disponibilidad no son características de prioridad para los análisis en la agricultura, tardar unos minutos en dar respuesta no afecta mucho la toma de decisiones, a menos que sean análisis muy detallados.

Apache Spark

Apache Spark [27] es un Framework de código abierto para procesamiento distribuido a gran escala, ofrece APIs de alto nivel para SQL, aprendizaje automático, Streaming, y procesamiento de grafos, disponibles en diferentes lenguajes como Python, Java, R, Scala. Adicionalmente, es agnóstico del sistema de almacenamiento y provee conectores junto con métodos para acceder a distintas fuentes de datos y combinarlas entre sí. Se han mostrado aplicaciones de Spark en un amplio rango de dominios desde finanzas hasta procesamiento de datos científicos, donde normalmente se combinan las librerías de alto de nivel. Desde su lanzamiento en 2010 Spark ha crecido hasta llegar a ser el proyecto código abierto más activo para procesamiento de Big Data. Spark unifica un conjunto de librerías de alto nivel, esto lo distingue de sistemas de procesamiento en clúster de propósito específico, como Storm, GraphLab e Impala, aun así compite con ellos en cuanto a rendimiento, para señalar el caso de uso publicado más grande es sobre un clúster de 8000 nodos de la red social China Tencent que atiende 1PB de datos por día [28].

Spark se considera la evolución del sistema de procesamiento de Hadoop debido a que lo supera en rendimiento (10x-100x) sumado a que extiende las características de map-reduce, se agregan nuevas funcionalidades, se destaca la posibilidad de persistir (caching) RDDs en Pipelines, lo cual evita múltiples lecturas en disco ante la ejecución de funciones iterativas que acceden a los mismo registros, contrario a map-reduce que accede a disco en todos los casos, causando una penalidad significativa en el rendimiento. Spark conserva la misma expresividad, escalabilidad, y tolerancia a fallos de la implementación de map-reduce con un Framework capaz de atender un mayor rango de aplicaciones [27].

Los programas en Spark se escriben en términos de dos tipos de operaciones: transformaciones y acciones, estas se ejecutan sobre conjuntos de datos distribuidos y optimizados para procesamiento en memoria. La estructura raíz de Spark se basa en la arquitectura sobre los Resilient Distributed Datasets (RDD) esta estructura es una abstracción para distribuir grandes conjuntos de datos entre nodos de un clúster y realizar operaciones sobre ellos tales como: filter, map, reduce, groupBy, etc... [27].

La implementación de Apache Spark se realizó en Scala, un lenguaje de programación de alto nivel estáticamente tipado para Java VM, Spark expone una interfaz de programación funcional similar a DryadLINQ. Los RDDs son objetos en Scala, inmutables, efímeros (por defecto), y de solo lectura [27]. Los RDDs soportan tolerancia a fallos con el concepto de Lineage, si una partición de un RDD se pierde, el RDD tiene suficiente información sobre cómo este fue derivado de otros RDDs y es capaz de reconstruir específicamente esa partición [28]

Resultados

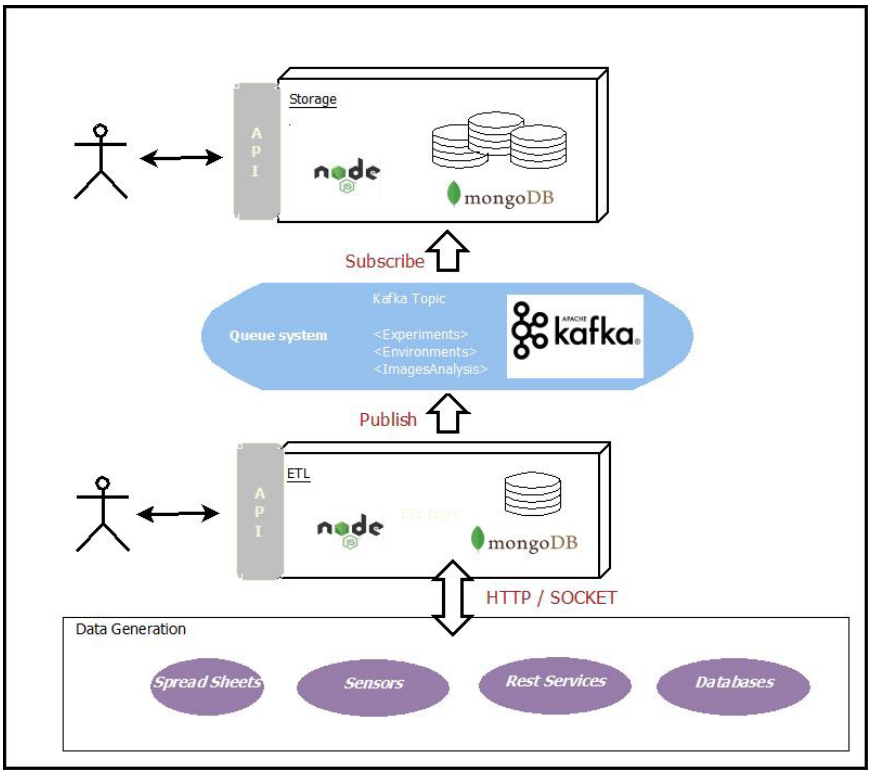
Arquitectura para el almacenamiento

Esta arquitectura se ocupa de los requerimientos R1-R4. En la Figura 2 se observa la arquitectura para el sistema de almacenamiento. En una visión de abajo hacia arriba se tienen los siguientes componentes:

- Fuentes de datos externas, estas se consumen o se acceden desde el ETL.
- Un ETL se implementa con la idea de microservicios, por estar desacoplado solo puede emitir mensajes al sistema de encolamiento con el método publish. La arquitectura propone el ETL como un componente de desarrollo manual para permitir conectar diferentes fuentes de datos, cumpliendo el requerimiento R2.

- El sistema de encolamiento es un mediador, evita que el clúster de almacenamiento deba recibir muchas conexiones, este sistema protege al almacén y propone un modelo de intercambio de mensajes claro y simple para la conexión de nuevas ETLs, se observa la etiqueta Kafka topics, cada tópico es un puente que define a qué colección pertenecen los datos.
- En el clúster almacén quedan guardados los datos, esta es una configuración de MongoDB que dependerá de los recursos disponibles, MongoDB garantiza escalabilidad pedida por el requerimiento R3.
- Finalmente se definen unas APIs de acceso a nivel del ETL para definir qué datos enviar al almacén y otro a nivel del almacén para visualizar los datos allí guardados.

Figura 2. Sistema de almacenamiento: (1) generación de datos se da por agentes externos, (2) la recolección de datos y validación, (3) el sistema de encolamiento, (4) el clúster escalable para almacenar grandes volúmenes de datos heterogéneos, y (5) el punto de acceso a través de web (APIs)



Fuente: Elaboración propia

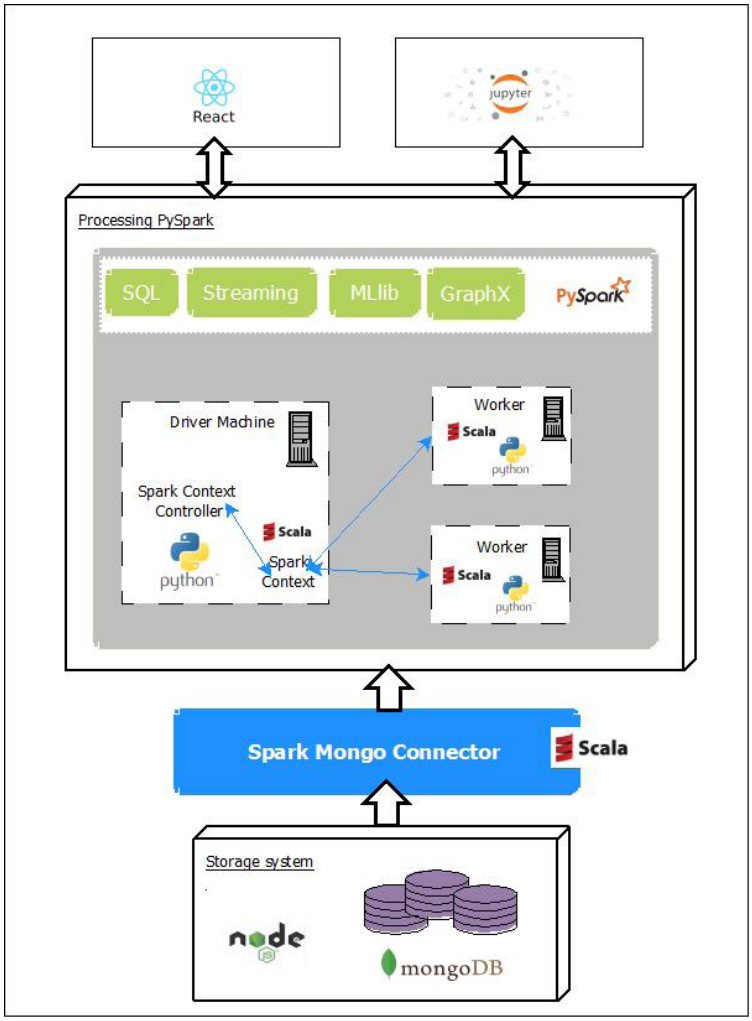
Arquitectura para el procesamiento

Esta arquitectura complementa los requerimientos mencionados R5-R8. En la Figura 3 se observa una visión general de la arquitectura, la base de la arquitectura son las tecnologías PySpark y MongoDB.

- El sistema de almacenamiento contiene los datos base que conectan con el sistema de procesamiento.
- La conexión se logra a través de un controlador de código abierto que establece un puente entre MongoDB y Spark, va de la mano con el requerimiento R4, este conector permite convertir una colección de MongoDB en un Dataframe de Spark. Entre las configuraciones para extraer los datos se puede definir un Pipeline de agregaciones que se ejecuta nativamente en MongoDB.

- PySpark se compone de un nodo driver y varios workers, en el driver se define un flujo que representa la aplicación y se hace un llamado de funciones en paralelo que serán ejecutadas en los workers en demanda, las funciones propias de Spark se ejecutarán en la Java VM por lo cual presentarán un mayor rendimiento, esto garantiza escalabilidad del procesamiento mencionada en el requerimiento R8.
- Se muestra en la arquitectura las librerías de más alto nivel de Spark, como lo son SQL, Streaming, MLib y graphX, en una sesión de Spark se pueden combinar llamados de estas librerías, para la creación de aplicaciones interactivas, van de la mano con los requerimientos R5-R6.
- Finalmente, la capa de acceso muestra que se pueden crear Notebooks o se puede acceder vía aplicación web y realizar invocaciones de manera interactiva.

Figura 3. Sistema de procesamiento: (1) base de datos MongoDB; (2) conector encargado de la comunicación; (3) Componentes lógicos de Spark, nodo driver y workers; (4) librerías de alto nivel de Spark: Mlib para aprendizaje automático, SQL para recuperación y agregación de datos, Streaming para procesamiento de flujos, y Graphx para de análisis de grafos; (5) interacción con usuarios a través de una aplicación web y de un Notebook



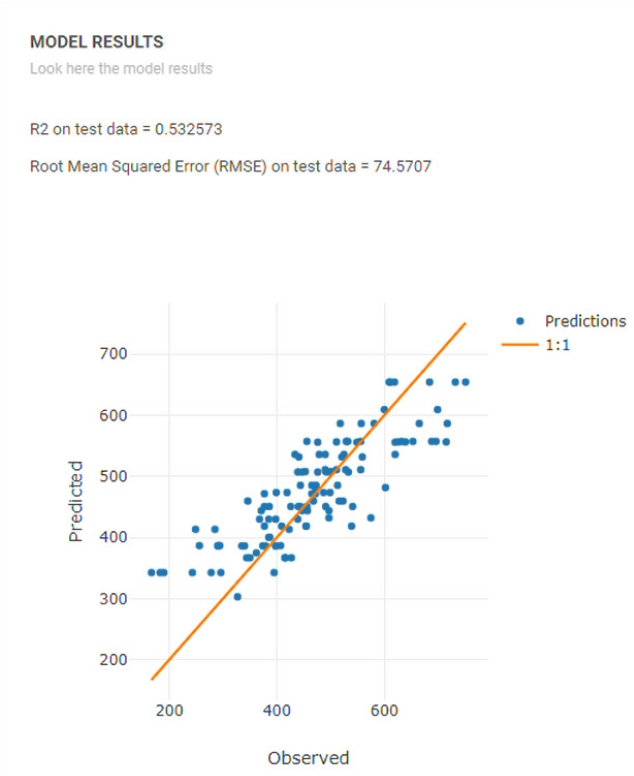
Fuente: Elaboración propia

Estos dos esquemas se complementan abarcando las necesidades expresadas en los requerimientos R1-R8, esta arquitectura tiene el potencial para crear un sistema robusto y servir de alternativa para hacer análisis de datos agronómicos heredando todas las bondades de las tecnologías de Big Data.

Prueba de concepto con datos de PHIS

Para validar los componentes se realizó una prueba de concepto. En esta se materializó la arquitectura diseñada a partir de componentes de software. Los datos de prueba se obtuvieron de la plataforma de fenotipado Phenoarch [29]. Esta plataforma genera datos diarios sobre el crecimiento de la planta. La arquitectura se adaptó a los procesos de obtención, almacenamiento y armonización de datos. Los datos que persisten en el sistema de almacenamiento se pueden consultar de manera interactiva para la ejecución de Pipelines de procesamiento y de procesos de análisis de datos sobre Spark. En esta prueba se ejecutó un modelo RandomForest de Spark para predecir la biomasa de la planta, este valor se puede considerar una medida de rendimiento a escala de planta. En la Figura 4 se presenta la vista previa de la prueba de concepto realizada.

Figura 4. Vista previa del resultado de la prueba de concepto



Fuente: Elaboración propia

Conclusiones

Para el análisis de datos en la agricultura se deben incorporar sistemas de almacenamiento flexibles y escalables. Al caracterizar los experimentos de predicción de rendimientos de cultivo se evidenció que los datos de entrada no son constantes y que contienen distintos niveles de detalle. El sistema propuesto es capaz de abarcar el almacenamiento de estos datos permitiendo la inclusión de variables de dominio específico y variables de series de tiempo. Al ser un sistema flexible permite centralizar datos heterogéneos, a la vez que impone un modelo de datos que facilita la integración de los experimentos.

Para obtener nuevos hallazgos al procesar datos de agricultura se deben incluir tecnologías modernas de Big Data sobre todo para manejar la variedad de los datos y para juntar experimentos de distintos estudios. El volumen aun no es una de las mayores preocupaciones en este dominio, sin embargo, al sobrepasar los problemas de integración estos retos aparecerán. Spark demostró ser una herramienta robusta que unifica análisis de datos con preprocesamiento, fue agnóstico para adaptar una base documental. Con esta combinación de tecnologías Spark-MongoDB-Kafka se cubren en gran parte los requerimientos presentados R1-R8. El valor agregado de utilizar estas herramientas es la capacidad de almacenar datos heterogéneos, y de distribuir el procesamiento de grandes volúmenes de datos.

Referencias Bibliograficas

1. Y. Everingham, J. Sexton, D. Skocaj, and G. Inman-Bamber, "Accurate prediction of sugarcane yield using a random forest algorithm," *Agronomy for Sustainable Development*, vol. 36, no. 2, pp. 1–9, 2016. [Online]. DOI: https://doi.org/10.1007/s13593-016-0364-z
2. D. Jiménez, J. Cock, H. F. Satizábal, A. Pérez-Uribe, A. Jarvis, P. Van Damme et al., "Analysis of andean blackberry (rubus glaucus) production models obtained by means of artificial neural networks exploiting information collected by small-scale growers in colombia and publicly available meteorological data," *Computers and electronics in agriculture*, vol. 69, no. 2, pp. 198–208, 2009. [Online]. DOI: https://doi.org/10.1016/j.compag.2009.08.008
3. A. Gonzalez-Sanchez, J. Frausto-Solis, and W. Ojeda-Bustamante, "Predictive ability of machine learning methods for massive crop yield prediction," *Spanish Journal of Agricultural Research*, vol. 12, no. 2, pp. 313–328, 2014. [Online]. DOI: http://hdl.handle.net/20.500.12013/1927
4. G. Ruß, "Data mining of agricultural yield data: A comparison of regression models," *Industrial Conference on Data Mining. Springer,* 2009, pp. 24–37. [Online]. DOI: https://doi.org/10.1007/978-3-642-03067-3_3
5. R. Lokers, R. Knapen, S. Janssen, Y. van Randen, and J. Jansen, "Analysis of big data technologies for use in agro-environmental science," *Environmental Modelling & Software*, vol. 84, pp. 494–504, 2016. [Online]. DOI: https://doi.org/10.1016/j.envsoft.2016.07.017
6. S. Delerce, H. Dorado, A. Grillon, M. C. Rebolledo, S. D. Prager, V. H. Patiño, G. G. Varón, and D. Jiménez, "Assessing weather-yield relationships in rice at local scale using data mining approaches," *PloS one*, vol. 11, no. 8, p. e0161620, 2016. [Online]. DOI: https://dx.doi.org/10.1371%2Fjournal.pone.0161620
7. L. Wasserman, *All of statistics: a concise course in statistical inference.* Springer Science & Business Media, 2013.

8. D. Jiménez, J. Cock, A. Jarvis, J. Garcia, H. F. Satizábal, P. Van Damme, A. Pérez-Uribe, and M. A. Barreto-Sanz, "Interpretation of commercial production information: A case study of lulo (solanum quitoense), an under-researched andean fruit," *Agricultural Systems*, vol. 104, no. 3, pp. 258–270, 2011. [Online]. DOI: https://doi.org/10.1016/j.agsy.2010.10.004
9. S. Haykin and N. Network, *A comprehensive foundation*, 2004, vol. 2, no. 2004.
10. G. Ruß, R. Kruse, M. Schneider, and P. Wagner, "Estimation of neural network parameters for wheat yield prediction," *IFIP International Conference on Artificial Intelligence in Theory and Practice*. Springer, 2008, pp. 109–118. [Online]. DOI: https://doi.org/10.1007/978-0-387-09695-7_11
11. J. R. Quinlan et al., "Learning with continuous classes," *5th Australian joint conference on artificial intelligence*, vol. 92. Singapore, 1992, pp. 343–348. [Online]. DOI: https://doi.org/10.1142/9789814536271
12. B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152. [Online]. DOI: https://doi.org/10.1145/130385.130401
13. S. R. Gunn et al., "Support vector machines for classification and regression," *ISIS technical report*, vol. 14, 1998.
14. L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
15. J. L. Riquelme, F. Soto, J. Suardáz, P. Sánchez, A. Iborra, and J. Vera, "Wireless sensor networks for precision horticulture in southern spain," *Computers and electronics in agriculture*, vol. 68, no. 1, pp. 25–35, 2009. [Online]. DOI: https://doi.org/10.1016/j.compag.2009.04.006
16. V. M. Ngo, N.-A. Le-Khac, M. Kechadi et al., "An efficient data warehouse for crop yield prediction," *arXiv preprint arXiv*:1807.00035, 2018. [Online]. DOI: https://arxiv.org/abs/1807.00035
17. A. Manjula and G. Narsimha, "Xcypf: A flexible and extensible framework for agricultural crop yield prediction," in *Intelligent Systems and Control* (ISCO), 2015 IEEE 9th International Conference on. IEEE, 2015, pp. 1–5. [Online]. DOI: https://doi.org/10.1109/ISCO.2015.7282311
18. A. Kamilaris, A. Assumpcio, A. B. Blasi, M. Torrellas, and F. X. Prenafeta-Boldú, "Estimating the environmental impact of agriculture by means of geospatial and big data analysis: The case of catalonia," *From Science to Society*. Springer, 2018, pp. 39–48. [Online]. DOI: https://doi.org/10.1007/978-3-319-65687-8_4
19. C. Bazzi, E. Jasse, E. Souza, P. S. Graziano MagalhÃ£es, G. Michelon, K. Schenatto, and A. Gavioli, "Agdatabox-api (application programming interface) a paper from the proceedings of the 14 th international conference on precision agriculture," 07 2018.
20. M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207–2219, 2016. [Online]. DOI: https://doi.org/10.1109/JPROC.2016.2598228
21. J. Mintert, D. Widmar, M. Langemeier, M. Boehlje, B. Erickson et al., "The challenges of precision agriculture: is big data the answer," in *Southern Agricultural Economics Association Annual Meeting,* San Antonio, Texas, no. 230057, 2016. [Online]. DOI: http://dx.doi.org/10.22004/ag.econ.230057
22. C. Rosenzweig, J. W. Jones, J. L. Hatfield, A. C. Ruane, K. J. Boote, P. Thorburn, J. M. Antle, G. C. Nelson, C. Porter, S. Janssen et al., "The agricultural model intercomparison and improvement project (agmip): protocols and pilot studies," *Agricultural and Forest Meteorology,* vol. 170, pp. 166–182, 2013. [Online]. DOI: https://doi.org/10.1016/j.agrformet.2012.09.011

23. J. W. White, L. Hunt, K. J. Boote, J. W. Jones, J. Koo, S. Kim, C. H. Porter, P. W. Wilkens, and G. Hoogenboom, "Integrated description of agricultural field experiments and production: The icasa version 2.0 data standards," *Computers and Electronics in Agriculture*, vol. 96, pp. 1–12, 2013. [Online]. DOI: https://doi.org/10.1016/j.compag.2013.04.003

24. I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice architecture: aligning principles, practices, and culture*. " O'Reilly Media, Inc.", 2016.

25. D. G. Chandra, "Base analysis of nosql database," *Future Generation Computer Systems*, vol. 52, pp. 13–21, 2015. [Online]. DOI: https://doi.org/10.1016/j.future.2015.05.003

26. N. Q. Mehmood, R. Culmone, and L. Mostarda, "Modeling temporal aspects of sensor data for mongodb nosql database," *Journal of Big Data*, vol. 4, no. 1, p. 8, 2017. [Online]. DOI: https://doi.org/10.1186/s40537-017-0068-5

27. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets." *HotCloud*, vol. 10, no. 10-10, p. 95, 2010.

28. M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin et al., "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016. [Online]. DOI: https://doi.org/10.1145/2934664

29. P. Neveu, A. Tireau, N. Hilgert, V. Nègre, J. Mineau-Cesari, N. Brichet, R. Chapuis, I. Sanchez, C. Pommier, B. Charnomordic et al., "Dealing with multi-source and multi-scale information in plant phenomics: the ontology-driven phenotyping hybrid information system," *New Phytologist*, vol. 221, no. 1, pp. 588–601, 2019. [Online]. DOI: https://doi.org/10.1111/nph.15385