Big Data Knowledge Acquisition Platform for Smart Farming

Van-Quyet Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
quyetict@utehy.edu.vn

Van-Hau Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
haunv@utehy.edu.vn

Minh-Quy Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
quynm@utehy.edu.vn

Quyet-Thang Huynh
School of Information and Communication Technology
Hanoi University of Science and Technology
Ha Noi, Vietnam
thanghq@soict.hust.edu.vn

Kyungbaek Kim
Department of Artificial Intelligence Convergence
Chonnam National University
Gwangju, South Korea
kyungbaekkim@jnu.ac.kr

ABSTRACT

Nowadays, big data enables to discover many aspects in agriculture sector such as finding unknown crop patterns or predicting the price of products. However, these massive data are often complex and heterogeneous which includes both structured (e.g., farm information) and unstructured data (e.g., image data, sensor data). It is required new techniques and tools to extract and represent valuable information in the form of human understanding to improve decision making for enhancing farm management. In this paper, we propose a big data knowledge acquisition platform which consists of efficient knowledge acquisition techniques integrated with an intuitive visualization tool supporting decision making applications. Firstly, we deploy open source big data frameworks (e.g., Flume, Hive, HBase) to support developing of multiple methods for collecting and storing data. Secondly, we implement distributed machine learning techniques on Hadoop and Spark to acquire knowledge from big data sources. Finally, we provide a visualization tool on web interface which can display extracted knowledge in multiple views (e.g., charts, tables) to support decision making applications. Experiments with real datasets show that the proposed platform is efficient and effective to answer important questions in smart farming.

CCS CONCEPTS

• **Computing methodologies → Distributed computing methodologies**; • **Information systems → Data analytics**.

KEYWORDS

Smart Farming, Big Data Platform, Knowledge Acquisition, Visualization, Decision Making System

ACM Reference Format:
Van-Quyet Nguyen, Van-Hau Nguyen, Minh-Quy Nguyen, Quyet-Thang Huynh, and Kyungbaek Kim. 2022. Big Data Knowledge Acquisition Platform for Smart Farming. In *The 11th International Symposium on Information and Communication Technology (SoICT 2022), December 1–3, 2022, Hanoi, Vietnam.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3568562.3568667

1 INTRODUCTION

Big data refers to the data sets with characteristics (4Vs): volume, velocity, variety, and value [3]. It comes from many areas in our life such as public health, social networks, and agriculture. For example, in agriculture domain, a huge amount of data is generated from many kinds of devices (e.g., temperature sensors, soil sensors, robotic drones) or agriculture services (e.g., weather forecast, market price). Besides, farmers can generate data during their growing products or manage their income by using traditional systems on a relational database. Thus, agricultural data become more and more complex with heterogeneity data structures (e.g., structured, semi-structured, and unstructured). It is necessary to develop a big data platform that utilizes state-of-the-art techniques and technologies for smart farming with respect to storage, capture, analysis, and visualization.

There are a number of studies focusing on data analytics and visualization to provide an understanding of the information contents for end users and experts. Based on Jsoup, Jie Wang et al. [23] has designed and implemented a platform for crawling and analyzing agricultural data. In which, the data are extracted from URLs of market websites, then these data are used for analyzing the influence of price changes and the market price trends of agricultural products to help consumers find cost-effective products from market information available. However, their platform only supported to deploy on a single computer which encounters of the challenges of big data problem. Chen et al. [5] suggested a crop breeding data analysis platform based on Hadoop/MapReduce [6]

SoICT 2022, December 1–3, 2022, Hanoi, Vietnam

ACM ISBN 978-1-4503-9725-4/22/12…$15.00
https://doi.org/10.1145/3568562.3568667

and Spark [25]. This system includes Hadoop distributed file system (HDFS) and cluster based on memory iterative components. A big data platform for collecting and analyzing agriculture has been proposed in [12], in which the authors presented multiple choices of each phase of handling big data. Nevertheless, this platform was lack of a big data visualization module to represent data in human understanding.

Big data visualization is a key part for creating an entire view and discovering values of data. Visualization techniques are used to create maps, tables, charts, and other forms to represent data. Most approaches and tools of previous data visualization are often inadequate to handle big data [1]. They are challenging to solve the drawbacks such as perceptual, real-time, and interactive scalability. In recent years, there are several researchers focusing on large-scale data visualization. Liu et al. [11] proposed imMens, which is a browser-based system using WebGL for data processing and rendering on the GPU. Some tools with functions of interaction for visualizing data are presented in [19]. Also, the extension of some conventional methods (e.g., Treemap, Streamgraph) for big data visualization is presented in [24].

In this paper, we propose a big data knowledge acquisition platform, consisting of efficient knowledge acquisition techniques integrated with an intuitive visualization tool supporting decision making applications. Firstly, we deploy open source big data frameworks (e.g., Flume, Storm, Sqoop, Hive, HBase) to support developing of multiple methods for collecting and storing data. Secondly, we implement distributed machine learning techniques on Hadoop and Spark to acquire knowledge from big data sources. Finally, we provide a visualization tool on web interface which can display extracted knowledge in multiple views (e.g., charts, tables) to support decision making applications. Through experiments with real datasets, we show that the proposed platform is efficient and effective to answer important questions in smart farming.

2 RELATED WORK

In modern agriculture, smart farming is being supported by advanced digital technologies [15], and big data frameworks [26]. Olakunle et al. [7] discussed several benefits and challenges of IoT and data analytics to the agriculture sector. The authors showed that using an IoT ecosystem with the support of data analytical subsystem empowers smart agriculture to deliver high operational efficiency with high productivity. Partha-Pratim presented a comprehensive review on IoT-based applications for smart agriculture [15], in which the author also proposed an IoT-based agricultural framework that takes full advantages of IoT for agriculture.

In recent years, a lot of studies have been focused on developing big data platforms that can handle massive volumes of agriculture data to support decision-making systems [26]. Shah et al. [17] proposed a spark-based agricultural information system built upon big data open sources. The author developed various web based analytical and visualized services for cotton crop. Tesfaye et al. [21] used geospatial and crop-modeling tools for analyzing big datasets in order to characterize the drought prevalence of countries in Southern Africa. Rajeswari et al. [14] proposed a smart agricultural model which integrates IoT, mobile and cloud databases into one a big data system.

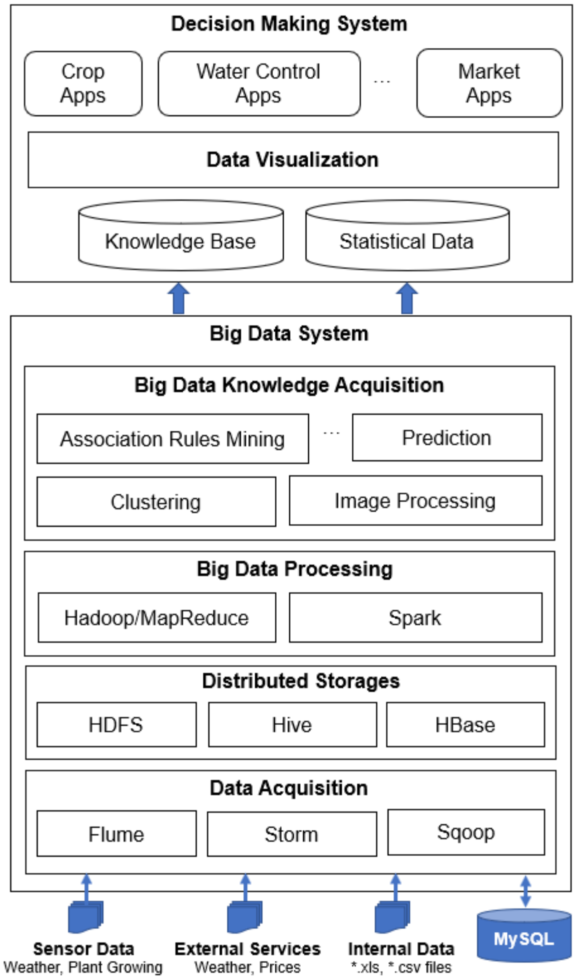


Figure 1: Architecture of the proposed platform

Despite many studies have focused on using image processing techniques in many agriculture areas such as plant pest detection and fruit grading [22], there has been few research studying on distributed image processing techniques or frameworks to deal with big data challenges in agriculture. To the best of our knowledge, there are three main frameworks that designed for image processing in Hadoop: HIPI (Hadoop Image Processing Interface) [20], OpenIMAJ (Open Intelligent Multimedia Analysis for Java) [8], and MIPr (Mapreduce Image Processing) [18]. However, these frameworks are required to modify the image storage such as HIP files in HIPI framework, which creates additional overhead in programming. In our work, we obtain automatically the images from the remote sensors and store them in HDFS without additional programming overhead for users to handle image storage.

3 A BIG DATA KNOWLEDGE ACQUISITION PLATFORM FOR SMART FARMING

We propose a platform for knowledge acquisition from agricultural big data which consists of (1) a big data system for analyzing data to extract knowledge from big data sources, and (2) a decision making system with intuitive visualization support. Figure 1 shows the architecture overview of our platform. The main components and functions of the systems are described as follows.

3.1 Big Data System

The proposed big data system aims to extract knowledge from massive and heterogeneous agricultural data. The system consists of four modules: Data Acquisition, Distributed Storages, Big Data Processing, and Big Data Knowledge Acquisition.

Data Acquisition. As a feature of big data, the veracity of the data is reflected through a variety of information sources. So, files held in office automation formats (e.g., *.csv, *.txt) scraped by crawling from the web, and other data sources that mentioned above need to be mixed-up as a basic raw data. It is not difficult to find that, such data can not be handled by a single personal computer due to the limitation of memory. Therefore, we propose a combination number of data acquisition techniques and tools to collect those kinds of data.

We separate agricultural data into two kinds of input, the first one is real-time data from web pages (e.g., weather data, market price) and the second one is historical data from archives (*.csv and *.xls files). For real-time data, we use Flume and Storm to collect data into HDFS. For historical data, usually with huge volume, we use Sqoop to import data into HDFS. Also, other data, such as the output data after analyzing, can be imported to MySQL by using Sqoop to provide for other systems/applications.

Distributed Storages. We use Hadoop Distributed File System (HDFS) as a basic big data distributed storage running on commodity servers with low-cost hardware. To provide quick random access to huge amounts of structured data, we use HBase which is a distributed column-oriented database built on top of HDFS. Moreover, to support SQL commands, a common type of data analysis, we use Hive, which is a data warehouse infrastructure tool to process structured data in Hadoop.

Big Data Processing. This module is an important component which performs algorithms and queries corresponding to the business tasks and user requests. To provide the best performance for extracting knowledge to support real-time decision making system, we use Spark and Hadoop/MapReduce engines for analyzing a large amount of data with various methods.

Big Data Knowledge Acquisition. To extract knowledge from massive and heterogeneous data, we proposed distributed machine learning and data mining techniques (e.g., clustering, prediction, association rules mining) that can be applied to various areas in agriculture. Specifically, we implement K-mean algorithm using MapReduce for solving big data clustering problem. Then, a SON-based algorithm using MapReduce is proposed for association rules mining from big data. We also propose a distributed image processing component, which supports to use Hadoop/MapReduce for handling a large number of images generated by cameras in smart farms. Furthermore, We implement VAR (Vector AutoRegressive) model on Spark by following the work presented by Tao et al. [10], for fast prediction deal with high-dimensional of the time series data in agriculture. In the future, other knowledge acquisition techniques such as deep learning based models (e.g., Convolutional Neural Network, Long Short-Term Memory) for solving classification problem and graph based techniques for extracting relationships between the entities (e.g., farm, location, devices) in agricultural data could be added to this module in order to provide more efficient and effective knowledge acquisition system.

3.2 Decision Making System

Knowledge Base. After extracting knowledge from big data sources, *Big Data Knowledge Acquisition* module will store the knowledge in *Knowledge Base.* There are two major kinds of knowledge bases, including human readable and machine readable. The former one enables people to access and use knowledge. It stores the information in the form of rules, documents, manuals which provide information users to guide the process and answer the frequently asked questions. While the later one stores the knowledge in the system readable forms. For instance, the output of using an image processing technique of detecting pests or weed from an image could be a text file which contains a matrix of binary values representing whether pest/weed exists or not in the locations.

Statistical Data. This data is also one kind of knowledge obtained from the results of Spark-based or Hadoop-based programs in data analysis. It could be the results of one time data processing without using machine learning techniques or data mining techniques. The statistical data take benefits of quick response to some questions requesting to unusual change data. For instance, to generate a report to answer the question: *who are the top 100 farmers obtaining the highest of income from growing onion*, we might need to join information from like farmers, products, and incomes. This work might take long response time due to the joining of *big tables.* To solve such a problem, we use Hive queries on MapReduce or Spark to process data and store the result into *Statistical Data* for later answering the question at other times.

Data Visualization. This is a critical module in a decision making system. It creates multiple views to help people to understand the data, from which people can make right decisions in their business tasks. To do this, we provide a big data visualization module with performance scalability and interactive scalability supports. This module supports to represent not only static knowledge stored in *Knowledge Base* and *Statistical Data* but also dynamic query that provide by users/experts via a web interface. The data is displayed in different styles: charts (e.g., bar chart, pie chart, line chart) and tables.

Applications. We consider this module as our future development for smart farming. It includes mobile, desktop, and web applications which support various areas in smart farming such as crop planning, water controlling, products selling in agriculture markets, etc.

4 IMPLEMENTATION OF BIG DATA KNOWLEDGE ACQUISITION

This part presents 4 modules: Data Clustering, Association Rules Mining, Image Processing, and Predicting.

4.1 Data Clustering Module

For clustering a huge amount of data in agriculture, we prefer to use an iterative algorithm such as K-means algorithm. It is one of the

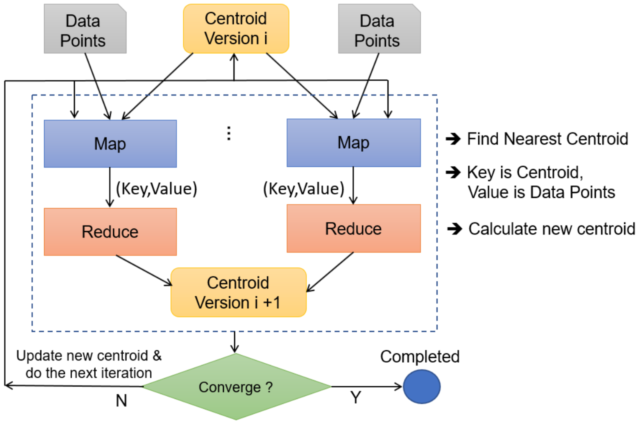


Figure 2: Parallel K-Mean algorithm on Hadoop/Spark

well-known machine learning techniques for clustering problem. The basic idea of this algorithm is that it groups objects (data points) based on features into k number of groups. Firstly, k data points are randomly selected as cluster centroids for k groups. Secondly, the algorithm performs calculating distances from every data points to each cluster centroid and assigns data points to clusters according to their distances. Thirdly, each cluster recomputes its cluster centroid based on its new assigned data points. Then, the algorithm performs the second step in some loops until the process converges.

We observed that the computation cost is mainly made in the step second, which calculates distances from every data point to each centroid in each iteration. Therefore, we can separate the distance calculation step from the main algorithm and implement a map-reduce function for this step on MapReduce and Spark. The procedure of implementation Map/Reduce functions on big data system is shown in Figure 2.

4.2 Association Rules Mining Module

4.2.1 SON Algorithm. In association rules mining technique, two entities should be considered: a set of items $I = I_1, I_2 \ldots, I_n$ and a collection of transaction $T = T_1, T_2 \ldots, T_n$ with each of them contains (some) independent item(s). Mining the association rules includes 2 main tasks: *frequent itemsets finding* and *association rules finding*. The former aims to get all the set of items whose rate of appearance (*support*) is larger than a given threshold, which is called *minimum support* (*minsup*). To do so, Apriori algorithm has been widely used [2]; however, it is not suitable due to a large volume of data because of the substantial memory space and the long response time. Therefore, the SON algorithm [16] is proposed for implementing the Apriori algorithm on parallel environment.

4.2.2 Distributed SON-based Algorithm. As mentioned before, our method is conducted based on SON algorithm on the parallel computing environment with Hadoop. In order to conduct this work, there should be 2 phases of MapReduce to handle 2 passes of the SON algorithm.

Phase 1: Finding global frequent itemset candidates

In this phase, the overall gotten list of transactions is divided into many smaller non-overlapping parts, which are handled by some mappers to find the local frequent itemsets on each smaller part. This procedure's output has the form *(key, value)* whose *key* is the item's name and *value* is a constant (e.g, value = 0) because this phase is just used for finding the possible frequent candidate, so its number of appearances need not to be considered. After being processed by the reducer, the overall *global frequent itemset candidates* are gotten.

Phase 2: Finding global frequent itemsets

In this phase, the list of global frequent itemset candidates is employed, which is assigned for each mapper. After the map procedure, the appearances of each itemset of global frequent itemset candidates on each smaller part will be gotten first. Then, their overall appearances over the whole dataset will be achieved once the reduce procedure completed. Based on this overall achievement, the support of each global frequent itemset candidates can be calculated, then decides whether itemsets are frequent or not.

4.2.3 Finding Association Rules. Based on the found frequent itemsets, the rules will be obtained by calculating their *confidence*. To do this, we follow the work mentioned in [16] generating subset of each frequent itemsets by each level of subset length decreasing. If any rules containing (A) as antecedent and (L/A) as consequent, where L is the set of all items of a given rule, could not satisfy the minimum confidence requirement, all the rules containing a's subset A_{subset} as antecedent and (L/A_{subset}) as consequent will be discarded without considering.

4.3 Image Processing Module

We consider the implementation of multiple variations of widely-used current image processing algorithms which are essential for big data analysis in agriculture such as plant pest detection and fruit grading. In order to support extracting knowledge from image data on the big data system, we divided image processing operations into two levels: low-level and high-level image processing.

Low-level Image Processing Algorithms

The algorithms call the image pre-processing to process at the pixel level [13], namely the low-level image processing algorithms. The low-level image processing operators require images as the input, while they output either images or data. For example, contrast enhancement, noise reduction, and noise removal in an image are low-level image processing operators. They are used for edge detection and diverse image transformations or calculation of simple characteristics, e.g. contours histograms.

High-level Image Processing Algorithms

The operations operating to generate higher abstractions [4] are called high-level image processing algorithms. They deal with abstractions developed from intermediate-level image processing operators. The algorithms are used to clarify the image content, e.g. classification and object recognition.

4.4 Predicting Module

Predicting module is an interesting part of the proposed system, although it is very challenging. Predicting helps farmers to choose appropriate right plants (around 730 types of plants in our dataset) to grow, consequently lead to their income and productivity. In our dataset, we collect information from Chonnanam-do province in South Korea in which we have important features: moisture, rainfall, and temperature.

We demonstrate this module by implementing a model which support of predicting the productivity of plants. To do that, we implement vector autoregression (*VAR*) model [9]. The reasons why we chose VAR model are following: (1) Our data are multivariate time series and autoregression; (2) VAR model is an extremely successful and flexible model for analysing of multivariate time series; (3) VAR model provides good forecasting capabilities; and (4) VAR model is easy to use.

A *VAR* model is a multi-equation system where all the endogenous (dependent) variables as a linear function of only past values. The model consists of k variables in a $(k \times 1)$ vector of time series variables. We denote $Y_t = (y_{1,t}, y_{2,t}, ..., y_{k,t})'$ as such a variable, where, at time t, the observation $-y_{i,t}-$ is the i^{th} element. If the i^{th} variable is productivity, then the value of productivity at time t is $y_{i,t}$.

The basic p-lag vector autoregressive, denoted *VAR(p)*, model has the form

$$Y_t = c + A_1 Y_1 + A_2 Y_2 + ... + A_{t-p} Y_{t-p} + \epsilon_t \qquad (1)$$

where c is a k-vector of intercepts, A_1 is a $(k \times k)$-matrix coefficient and ϵ_t is a k-vector of white noise processes:

(1) $E(\epsilon_t) = 0$ — unobservable zero mean
(2) $E(\epsilon_t \epsilon_t^{'}) = \Omega$ — where Ω is a $k \times k$ matrix of contemporaneous covariance;
(3) $E(\epsilon_t \epsilon_{t-k}^{'}) = 0, \forall k \neq 0$

The simplest example is the vector autoregressive process with two variables ($k = 2$), VAR(1), as in

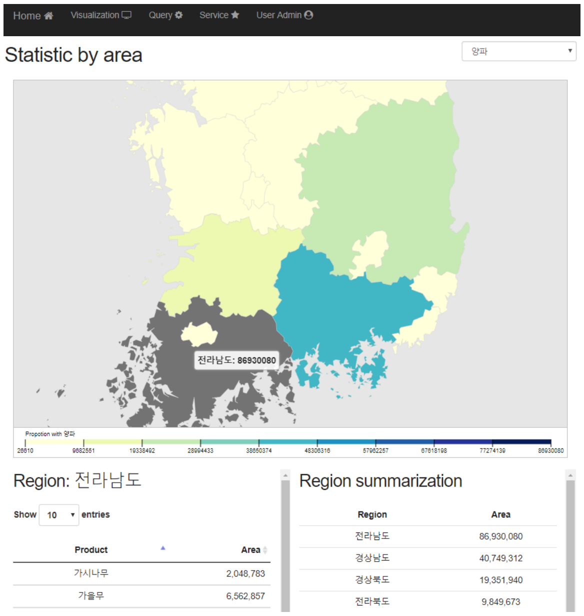
$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \end{bmatrix}$$

5 VISUALIZATION SUPPORT OF DECISION MAKING APPLICATION

In this section, we present a design and implementation of a big data visualization module that allows users/experts to explore agricultural big data. Firstly, we provide a client web interface, in which, in order to see data visualization, users make requests to the Web server. Secondly, the web server checks if the request is a static knowledge visualization, including both *Knowledge Base* and *Statistical Data*, then, it queries data on MySQL Server and instantly shows the result; otherwise, it sends the queries to Hive Server and waits to get dynamic data from Hive warehouse. Here, MapReduce or Spark engine could be used to handle data. Finally, once the web server gets the result from static or dynamic knowledge, it responds to the web client and show the result in a visual view (i.e., a bar chart, a line chart, or a table).

Next, we give a brief overview of the tasks that a user/expert can carry out using the data visualization module for decision making applications and describe how it was used in our case study.

Exploring growing plants by region. When agriculture researchers encounter the integrated data for the first time, one of the important tasks they want to carry out is taking a look at an overview of growing plants by regions. To accomplish that, we have created a region map (Korea Map in our case) as shown in Fig. 3. It provides a quick overview of the growing plants of the region, such as types of plants and planted area.



Statistic by area

Region: 전라남도

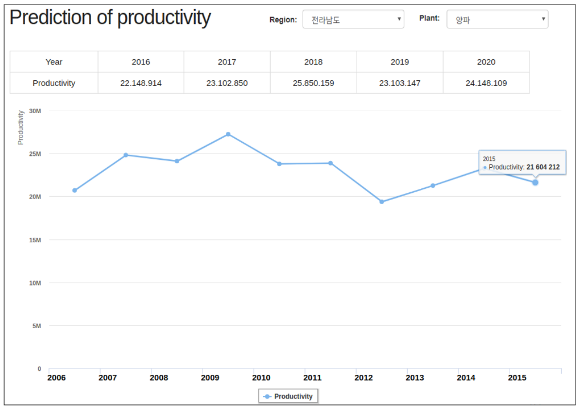
Show 10 entries

Product	Area
가시나무	2,048,783
가을무	6,562,857

Region summarization

Region	Area
전라남도	86,930,080
경상남도	40,749,312
경상북도	19,351,940
전라북도	9,849,673

Figure 3: Statistic of the grown plants with Korea map visualization



Prediction of productivity

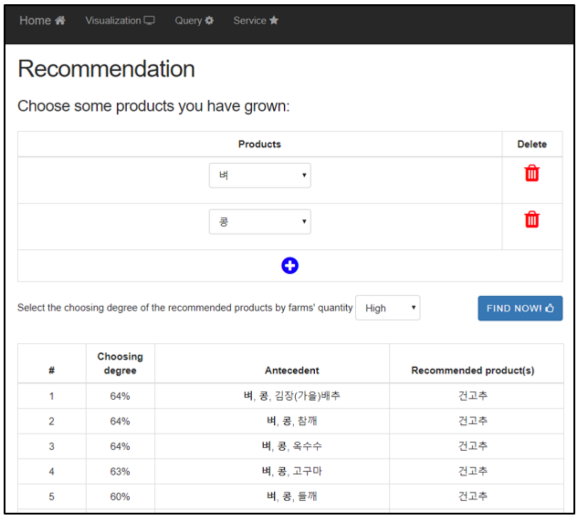
Region: 전라남도 Plant: 양파

Year	2016	2017	2018	2019	2020
Productivity	22.148.914	23.102.850	25.850.159	23.103.147	24.148.109

Figure 4: Visualization of predicting plant production

Predicting plant production. In fact, every farmer and agriculture manager would like to predict their plant production for the current season as well as the next seasons. To support that, we design a visualization module for predicting the production of each plant in each region. Also, farmers can forecast plant production on their farms. Figure 4 illustrates the prediction of onion productivity of Chonnanam-do province in South Korea.

Finding associations rules related to plants growing. Choosing which plants should be grown in a season is a very crucial question to farmers because it is a factor for manipulating their income and productivity. Usually, each farmer would choose a few main plants



Recommendation

Choose some products you have grown:

Products	Delete
벼	
콩	

Select the choosing degree of the recommended products by farms' quantity: High

#	Choosing degree	Antecedent	Recommended product(s)
1	64%	벼, 콩, 김장(가을)배추	건고추
2	64%	벼, 콩, 참깨	건고추
3	64%	벼, 콩, 옥수수	건고추
4	63%	벼, 콩, 고구마	건고추
5	60%	벼, 콩, 들깨	건고추

Figure 5: Visualization of recommended association rules related to plants growing

which they think those are most suitable for them. However, in order to get the most benefits, they should also choose other types to grow with their chosen ones. This is a big question because among many plants (e.g., around 730 types of plants in our dataset, see the next section for more details), finding the suitable one requires a lot of conditions such as high frequency growing by other farms and high production. To support farmers do so, we used distributed parallel association rule mining technique what is presented in the previous section. Then, the output store in Knowledge Bases in the form of rules that can be shown in the web interface as illustrated in Fig. 5.

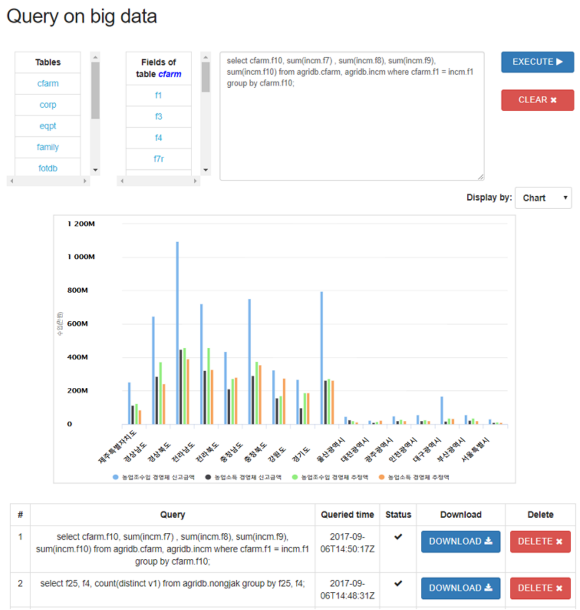
Exploring agricultural big data by using dynamic visualization with Hive query. For experts researching on agricultural big data, dynamic visualization is one of the best ways to discover great insight from big data. In order to do that, we developed a web-based visualization as shown in Fig. 6. In which, it enables users to make a Hive query, then send the query to Hive Server to execute the query by MapReduce and get the result. Finally, the result can be plotted in multiple views, such as chart, table, or file(s) depending on its size.

6 EVALUATION

6.1 Evaluation Settings

Big Data System. We conducted our evaluation with a Hadoop cluster on 5 machines: one for master node and 4 for compute nodes. Each machine has 4 CPU and 16 GB of RAM.

Dataset. To explore how agricultural big data can be leveraged to offer benefits for farmers/agribusiness, we worked on a case study of agriculture in South Korea. Three different datasets are integrated to discover interesting knowledge. The first dataset contains approximate 10GB of agricultural data of 16 regions in South Korea from 2015 to 2016, which includes information about farms, income, soil data, production, and other data. Two other datasets (weather data and market price data) were collected by our system.


Query on big data

Tables
cfarm
corp
eqpt
family
fotdb

Fields of table *cfarm*
f1
f3
f4
f7r

```
select cfarm.f10, sum(incm.f7) , sum(incm.f8), sum(incm.f9),
sum(incm.f10) from agridb.cfarm, agridb.incm where cfarm.f1 = incm.f1
group by cfarm.f10;
```

Display by: Chart

#	Query	Queried time	Status	Download	Delete
1	select cfarm.f10, sum(incm.f7) , sum(incm.f8), sum(incm.f9), sum(incm.f10) from agridb.cfarm, agridb.incm where cfarm.f1 = incm.f1 group by cfarm.f10;	2017-09-06T14:50:17Z	✓	DOWNLOAD	DELETE
2	select f25, f4, count(distinct v1) from agridb.nongjak group by f25, f4;	2017-09-06T14:48:31Z	✓	DOWNLOAD	DELETE

Figure 6: Visualization of dynamic querying on big data

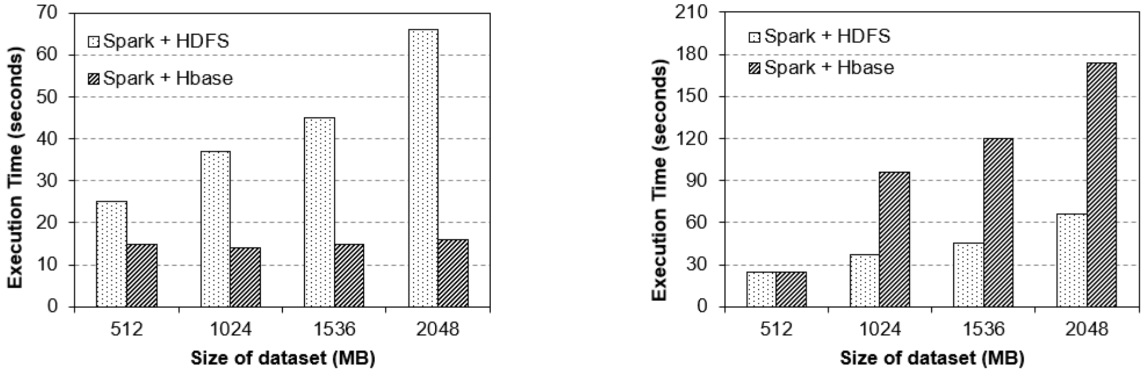
6.2 Experimental Results

Exp-1: Efficiency of distributed storages

In order to evaluate the performance of data storages in our platform, we set up an experiment on Spark to compare the execution time of searching data stored in HDFS and HBase. In the first case, we perform a sequential searching to calculate the total field area of each farm in the agricultural dataset (a table with 34 columns, 2GB data). It is useful for a manager who monitors the statistic information of each farm. Another case is random searching to calculate total field area of some given farms specified by Farm IDs. It is useful for a user to get the information of a specific farm. Figure 7 illustrates the results of performance comparison of the two cases above. In the sequential searching case (Fig. 7-(a)), Spark with HDFS performs faster than Spark with HBase. On the other hands, in the case of random searching (Fig. 7-(b)), Spark with HBase has a better performance than Spark with HDFS. Because HBase provides Column-Family mechanism and Row-ID access, Spark with HBase spends less time in reading data in random.

Exp-2: Efficiency of distributed clustering

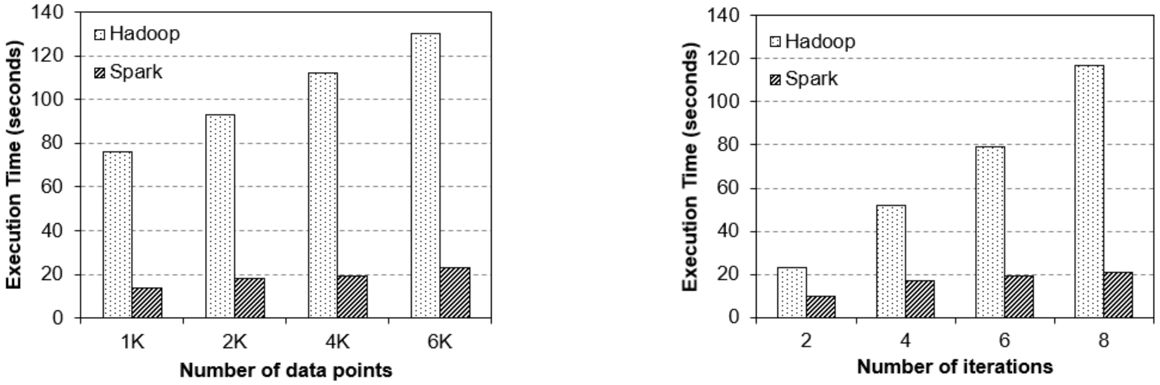
To evaluate the efficiency of our distributed clustering module in our big data knowledge platform, we performed clustering soil data based on its chemical characteristics (e.g., pH, KCl). We evaluated the performance of distributed clustering with both Hadoop and Spark on various size of data points (1K to 6K) and the number of iterations (2 to 8). Fig. 8 depicted the performance of group soil data into 4 clusters in case of the convergence is achieved. We observed that Spark is 5.5x faster than Hadoop with a various number of data points as shown in Fig. 8-(a), whereas Fig. 8-(b) shows that Spark



(a) Sequential Searching

(b) Random Searching

Figure 7: Evaluating the efficiency of different Distributed Storages



(a) Varied data points

(b) Different number of iterations

Figure 8: Evaluating the efficiency of Distributed Clustering module

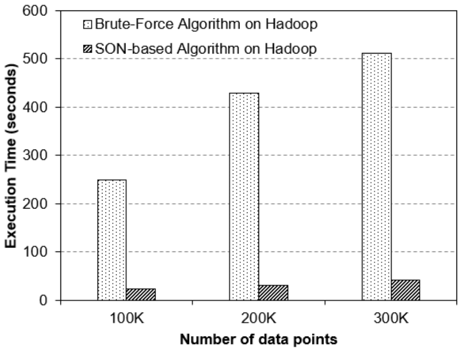


Figure 9: Evaluating the efficiency of Distributed Association Rules Mining module

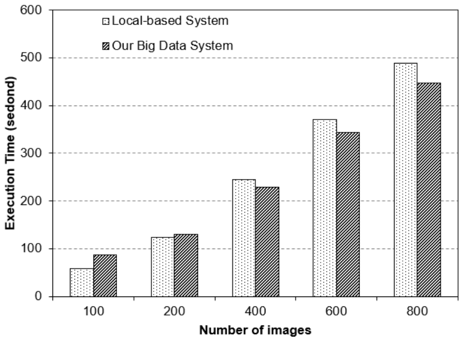


Figure 10: Evaluating the efficiency of Distributed Image Processing module

is more efficient than Hadoop when the number of iterations of the algorithm increases. Specifically, Spark is faster than Hadoop about 2 times after 2 iterations, and approximately 6 times after 8 iterations.

Exp-3: Efficiency of distributed association rule mining.

In this experiment, we use *farm dataset* containing information about the identity number, grown products, grown area, and income. Because a specific farm with a unique identity number could have many grown products, there could be many rows with the same identity number, but different grown product. Consequently, the identity number and grown products are chosen as transaction field and item field, respectively. Moreover, in order to increase the quality of the gotten rules, we only get the transactions from a farm whose income is larger than a given number (e.g., 10.000 thousand won). This filter proves that the gotten rules are retrieved from good sources. To evaluate the efficiency of our platform in finding association rules, we compared our SON-based distributed algorithm with a baseline mother called Brute-Force algorithm. Both of our method and the original method are conducted on proposed Hadoop environment. Figure 9 illustrated the performance of Brute-Force algorithm and our SON-based algorithm on Hadoop. From this figure, we observed that the execution time of our method is always much lower than the original method when the number of transactions increases.

Exp-4: Efficiency of distributed image processing.

To evaluate the performance of the image processing module, we tested our big data system with histogram calculation problem on a dataset of 800 images. The dataset contains crop/weed images which are collected by our platform from the Internet. To compare the performance of our system with a traditional image processing system, we also implemented a program in Java which conducts the same algorithms in a single machine, called local-based system. Figure 10 shows the comparison of the execution time between our proposed system and local-based system. We observed that with a small size of the image dataset, the execution time of local-based system is smaller than our system. Because our system needs to read/write the data at the initial state of the whole process, it spends more time to set up the process compared to the one on local-based system. However, with large size of image dataset (> 400 images), our system executes faster than local-based system. Thus, our system is more scalable in aspects of the volume of the input dataset.

7 CONCLUSIONS

This paper proposed a big data knowledge acquisition platform which consists of several efficient knowledge acquisition techniques integrated with an intuitive visualization tool supporting decision making applications. We first deployed open source big data frameworks to support developing of multiple methods for collecting and storing data. We then implemented distributed machine learning techniques on Hadoop and Spark to acquire the knowledge from big data sources. Finally, we provided a visualization tool on a web interface which can display extracted knowledge in multiple views (e.g., charts, tables) to support decision making applications. Through experiments with real datasets, we showed that our platform is efficient and effective to answer important questions in smart farming.

ACKNOWLEDGMENTS

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2022-RS-2022-00156287) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

REFERENCES

[1] Rajeev Agrawal, Anirudh Kadadi, Xiangfeng Dai, and Frederic Andres. 2015. Challenges and opportunities with big data visualization. In *Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems.* ACM, 169–173.

[2] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. 487–499.

[3] G Amir and H Murtaza. 2015. Big data concepts, methods and analytics. *International Journal of Information Management* 35 (2015), 140.

[4] Thomas Bräunl, Stefan Feyrer, Wolfgang Rapf, and Michael Reinhardt. 2013. *Parallel image processing.* Springer Science & Business Media.

[5] Shuangxi Chen, Chunming Wu, and Yongmao Yu. 2016. Analysis of plant breeding on hadoop and spark. *Advances in Agriculture* 2016 (2016).

[6] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.

[7] Olakunle Elijah, Tharek Abdul Rahman, Igbafe Orikumhi, Chee Yen Leow, and MHD Nour Hindia. 2018. An overview of Internet of things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal* 5, 5 (2018), 3758–3773.

[8] Jonathon S Hare, Sina Samangooei, and David P Dupplaw. 2011. OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *Proceedings of the 19th ACM international conference on Multimedia.* ACM, 691–694.

[9] Abdulnasser Hatemi-J. 2004. Multivariate tests for autocorrelation in the stable and unstable VAR models. *Economic Modelling* 21, 4 (July 2004), 661–683. https://ideas.repec.org/a/eee/ecmode/v21y2004i4p661-683.html

[10] Tao Li, Xueyu Li, and Xu Zhang. 2017. The Design and Implementation of Vector Autoregressive Model and Structural Vector Autoregressive Model Based on Spark. In *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM).* IEEE, 386–394.

[11] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. 2013. imMens: Real-time Visual Querying of Big Data. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 421–430.

[12] Van-Quyet Nguyen, Sinh Ngoc Nguyen, and Kyungbaek Kim. 2017. Design of a platform for collecting and analyzing agricultural big data. *Journal of Digital Contents Society* 18, 1 (2017), 149–158.

[13] Cristina Nicolescu and Pieter Jonker. 2000. Parallel low-level image processing on a distributed-memory system. In *International Parallel and Distributed Processing Symposium.* Springer, 226–233.

[14] S Rajeswari, K Suthendran, and K Rajakumar. 2017. A smart agricultural model by integrating IoT, mobile and cloud-based big data analytics. In *2017 International Conference on Intelligent Computing and Control (I2C2).* IEEE, 1–5.

[15] Partha Pratim Ray. 2017. Internet of things for smart agriculture: Technologies, practices and future direction. *Journal of Ambient Intelligence and Smart Environments* 9, 4 (2017), 395–420.

[16] Ashok Savasere, Edward Robert Omiecinski, and Shamkant B Navathe. 1995. *An efficient algorithm for mining association rules in large databases.* Technical Report. Georgia Institute of Technology.

[17] Purnima Shah, Deepak Hiremath, and Sanjay Chaudhary. 2017. Towards development of spark based agricultural information system including geo-spatial data. In *2017 IEEE International Conference on Big Data (Big Data).* IEEE, 3476–3481.

[18] Andrey Sozykin and Timofei Epanchintsev. 2015. MIPr-a framework for distributed image processing using Hadoop. In *2015 9th International Conference on Application of Information and Communication Technologies (AICT).* IEEE, 35–39.

[19] Venkataramana Sucharitha, SR Subash, and P Prakash. 2014. Visualization of big data: its tools and challenges. *International Journal of Applied Engineering Research* 9, 18 (2014), 5277–5290.

[20] Chris Sweeney, Liu Liu, Sean Arietta, and Jason Lawrence. 2011. HIPI: a Hadoop image processing interface for image-based mapreduce tasks. *Chris. university of Virginia* 2, 1 (2011), 1–5.

[21] Kindie Tesfaye, Kai Sonder, J Caims, Cosmos Magorokosho, Amsal Tarekegn, Girma T Kassie, Fite Getaneh, Tahirou Abdoulaye, Tsedeke Abate, and Olaf Erenstein. 2016. Targeting drought-tolerant maize varieties in southern Africa: a geospatial crop modeling approach using big data. (2016).

[22] Anup Vibhute and SK Bodhe. 2012. Applications of image processing in agriculture: a survey. *International Journal of Computer Applications* 52, 2 (2012).

[23] Jie Wang, Shuo Yang, Yuezhi Wang, and Cheng Han. 2015. The crawling and analysis of agricultural products big data based on Jsoup. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD).* IEEE, 1197–1202.

[24] Lidong Wang, Guanghui Wang, and Cheryl Ann Alexander. 2015. Big data and visualization: methods, challenges and technology progress. *Digital Technologies* 1, 1 (2015), 33–38.

[25] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster computing with working sets. *HotCloud* 10, 10-10 (2010), 95.

[26] Ji-chun Zhao and Jian-xin Guo. 2018. Big data analysis technology application in agricultural intelligence decision system. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA).* IEEE, 209–212.