Koi Fish Warehousing Based on Mobile Cloud Computing

Chao-Lieh Chen
Dept. Electronic Engineering
First Campus, National Kaohsiung University of Science and Technology
Kaohsiung, Taiwan
frederic@ieee.org

Chia-Chun Chang
Dept. Electronic Engineering
First Campus, National Kaohsiung University of Science and Technology
Kaohsiung, Taiwan
jamesrmp4@gmail.com

***Abstract*—We propose an application paradigm of mobile cloud computing with big video data for ornamental fish warehousing (OFWare) system dedicated to koi fishes. In fields of aquaculture and agriculture, live creatures are the main products and are difficult to perform warehouse management. Warehousing of high unit-price ornamental fishes such as koi, stingray, and arowana is even more difficult since the warehousing requires identification, in addition to simple counting and classification, of individuals whose shapes and texture patterns are time-variant as they grow. Therefore, rather than using invasive RFID-based systems, we combine mobile cloud computing and big data analytics techniques including image and video collecting and transmission by handheld mobile devices, unsupervised texture pattern classification in fish tank videos, image retrieval based on support vector machine, and statistical analysis for the warehouse management. The proposed system is scalable based on Hadoop framework and a small unit of a single name-node and data-node is able to identify a fish feature among 500,000 koi fishes in 7 seconds.**

Keywords—Aquaculture Business Intelligence, Image Retrieval, Mobile Cloud Computing, Machine Learning, Koi Fish Warehousing System.

I. INTRODUCTION

Intelligent aquaculture depends on technology application as is widely exploited in the Industrie 4.0 [1]. The ornamental fish industry is one of the most worth studying the application of information and communication technologies in aquaculture. Take the koi fish market as an example, though it is billion-dollar global market value, breeding techniques require many years of heritage and thus it is difficult to maintain a larger business scale. The basis of the business development is the warehousing system of koi fishes. However, it is difficult to identify koi individuals merely by a staff's eyes. It is possible to misplace one high/low price fish for a low/high price order because even an experienced staff would misidentify individuals. Traditional RFID-based systems require one or more staffs to catch each koi individual and then perform invasive RFID implanting. The process risks each fish life and is still time consuming. Rather than using invasive RFID for identification, we propose image retrieval-based system for warehouse management of ornamental fishes. When a staff takes video of a fish tank using only a handheld device camera, the system detects the number of fishes by unsupervised classification and then captures multiple fish individuals' images at one time. To enhance the system performance, we offload image-processing tasks along the network path from the handheld client to the cloud. The handheld device performs image resizing and human-aided segmentation. The RESTful operations [2] allows image and data access between the mobile device and the cloud server. The webserver performs image pre-processing such as morphological operations to extract each fish's texture features. The cloud server in Hadoop ecosystem performs the image retrieval for making inventory and breeding history update. The OFWare becomes the development platform for the following items.

(1) Inbound, outbound, breeding position change, and logistics: in these scenarios, it is essential to make inventories.

(2) Business analytics: this will include feed, medicine, feeding supplies, environment construction, family variety, pattern preferences global distribution, price distribution, customer and pattern preferences analysis, attention analysis, and customized reports as well as the relations among these items.

(3) Tutorial about breeding techniques: the breeding history includes texture patterns of individuals in addition to family variety and environmental sensor data. This shortens the time of heritage and enhanced health management of fishes.

(4) Certificates of owner, origin, and competition: anti-forgery is achieved by photos of fish individuals taken by visitors in different places and different time. The process of transportation in important logistics stops such as customs is recorded in the platform's database as well as in the staffs' handheld devices.

This paper is an extension of [3] where feature extraction for machine learning and big data ecosystem for cloud computing platform are detailed. To the authors' best knowledge, this is the first live creature warehousing system where shapes and values of "cargo" are time-variant. The rest of the paper is briefed as follows. Section II provides the system framework with message flows. Detail descriptions of key components such as image preprocessing, feature extraction example, and big data ecosystem construction are provided in Section III. Section IV is the system operation results. Finally, we give conclusion in Section V.

II. SYSTEM FRAMEWORK

Shown in Fig. 1 is the proposed OFWare system framework. There four subsystems, which are handheld device, image preprocessor, machine learning engines, and the Hadoop ecosystem. The messages labeled with sequence numbers between subsystems are highlighted with red color. We detail the message flows as follows.

1: The user provides settings including number of fishes in the temporary tank and select the machine learning algorithm for training and recognition.

2: The sequence of images is encoded in Base64 format for transmission across the Internet.

3: Through network equipment, the images and system settings are sent to the web browser, which undertakes the image preprocessing tasks including feature extraction and unsupervised feature classification.

4 and 5: The features of fishes are extracted. Appending with auto-generated IDs, the OFWare invokes the machine learning engines on the Hadoop framework.

6: The new features are used for query the images in the HDFS.

7.and 8: Since the fishes might grow with new texture changes, the new features becomes new training samples.

9-12: During training, the images and features are scattered for storing in the HDFS (message 9). When prediction, old machine learning model is used (message 10). At the same time, the new features are the new training samples for producing the machine learning model (message 11) and updating the model (message 12).

13-15: Finally, the OFWare send the image query results back to the mobile device also using node.js for rendering on the web browser.

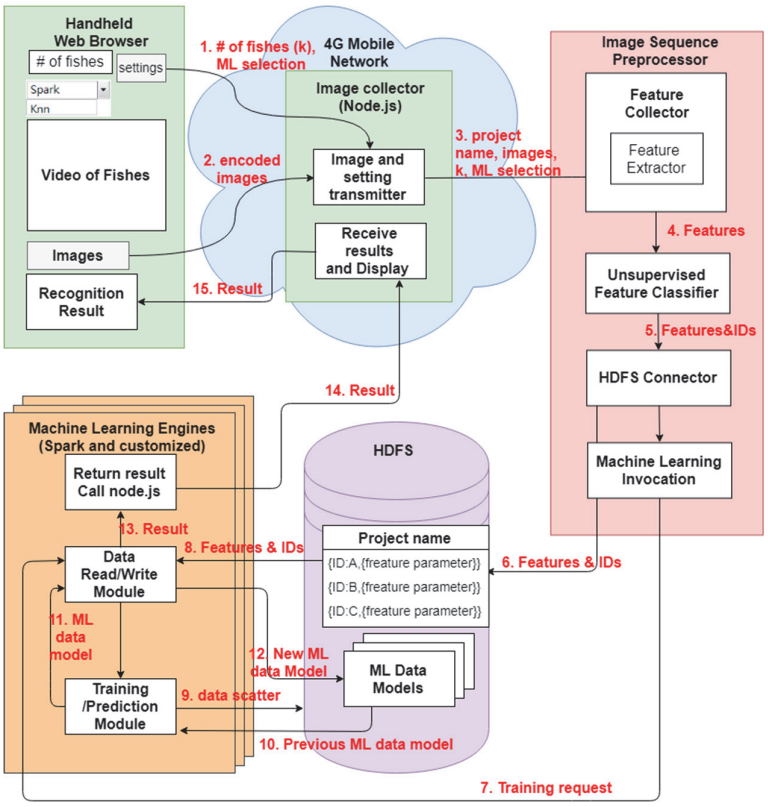

Fig. 1. Framework of the proposed ornamental fish warehousing system.

III. Key Components

A. Mobile device

Through web-based user interface, a user uses handheld camera to capture a fish video. The user gives the number of fishes in a temporary fish tank for transportation or making inventory. Advanced settings such as machine learning and related parameter options are also available for engineer users. After submitted the video, the OFWare system returns results of recognition and the user starts detail operations for each fish individual such as creating or updating fish individual's profile, inventory, reading visualized data, invoking accounting process, …, etc. Through the *node.js* library [4], the OFWare converts the fish video into image sequences and encodes the images for transmitting to the cloud.

B. Image Preprocessor

The OFWare is able to process multiple fishes at a time. Usually there are no more than 10 fishes in a temporary fish tank. Therefore, it is easy for a simple server, such as a webserver, to perform unsupervised classification of the fishes' features. The primary preprocessing techniques include background removing, warping, and fish color texture segmentation. Shown as Fig. 2, usually, the background of a fish tank image is blue. Therefore, applying color histogram thresholding, we obtain the fish close-ups. Once there are some occlusion conditions, the GUI on the mobile device provides blue pen for users to draw blue separation lines. This ease the preprocessing while not reduce quality of user experience. For video sequence input, since the number of fishes in the tank is known, the segmentation of color texture

patterns are based on unsupervised classification. The most dense clusters cover fishes' color textures never occluded.

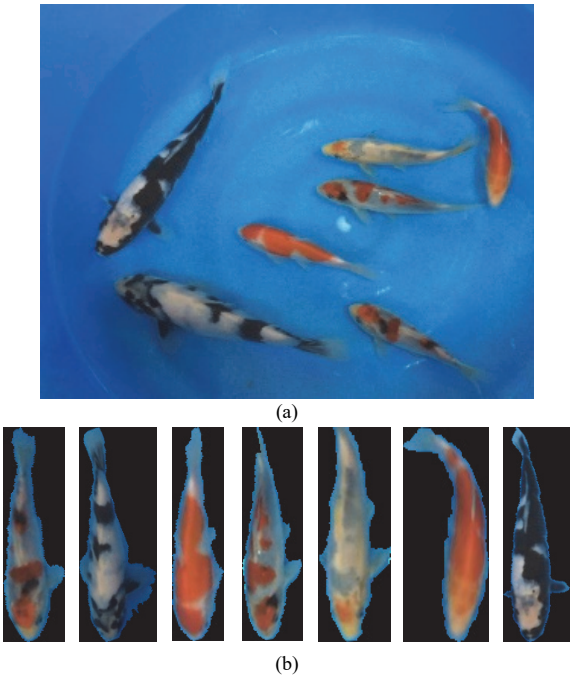

Fig. 2. Image segmentation and warping to obtain each fishes' straight close-ups. (a) input image frame Image segmentation; (b) fishes' close-ups.

C. *Feature Collector and Extractor*

Receiving the image sequence, the webserver segments each image into close-ups of individual fishes. Then, it extracts and collects the fishes' texture pattern features. To preserve the texture orientation, the feature extractor rotates and stretches individual close-ups to make each of the fish image upright before extracting the features of texture patterns. The features are based on LTP (local ternary patterns) [5]. Each fish's feature vector is a concatenation of LTP features of different color texture.

Here is an example for the LTP feature extraction. Suppose positive threshold for LTP is 3, negative threshold = -1, p(0), and p(i), i=0, 1, …, 7 are pixels in an 3x3 LTP sampling window centered at p(0). The encoding of orientation of the sampling window is

$$bp_i = \begin{cases} 1 & if\ p(i) - p(0) > 3 \\ 0 & else \end{cases}$$

$$bn_i = \begin{cases} 1 & if\ p(i) - p(0) < -1 \\ 0 & else \end{cases}$$

Therefore, the contributions of p(0) to the LTP histograms are respectively as follows

$$hp = \sum_{i=0}^{7} bp_i\, 2^i$$

$$hn = \sum_{i=0}^{7} bn_i\, 2^i$$

Fig. 3 is a fish image example where red and black texture patterns are separately extracted. Fig. 4 are the LTP histograms (feature vectors) for the fish image in Fig. 3. Shown in Fig. 4, red and black textures of this fish have respective LTP histograms with 256 bins from bin 0 to bin 255. They are concatenated as into a longer feature vector representing the fish.

Fig. 3. Fish texture pattern. (a) fish image with background removed; (b) red texture pattern of the fish; (c) black texture pattern of the fish.

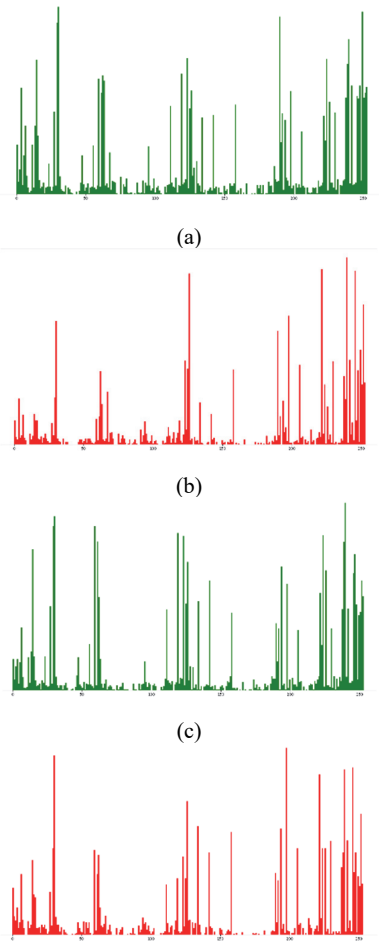

Fig. 4. LTP histograms of the fish picture in Fig. 3: (a) red texture pattern(positive); (b) red texture pattern(negative); (c) black texture pattern(positive); (d) black texture pattern(negative)

D. Unsupervised Feature Classifier

The unsupervised classifier classifies fish features in the sequence of images. Then, the OFWare is able to collect more features of a single fish in different situations such as in different lighting, shadowing, and view angles. Through the unsupervised feature classifier, Gaussian Mixture Model (GMM) , we obtain a more representative fish feature of each fish compared to that extracted from only a single image. The number of fishes in the temporary tank is then the number of classes given by the user in the web browser UI.

The unsupervised classifier is use to detect number of fishes in a temporary breeding tank. The recognition suffers from shadows, water wave reflections, fish overlapping, fish body occlusion and many other uncertain interferences. However, since in a temporary breeding tank the number of fishes and their texture patterns are fixed, the OFWare receives fishes' texture patterns (interfered or not) from continuous image frames in the video. The classifier in the OFWare then classifies the texture patters from the video. Since the user provide the number of fishes in a tank via the UI interface, the OFWare system chooses the most representative fish classes, which are condensed and with large member amount, for later identification. In a 10-second 30fps video of 5 fishes, there will be 300 frames and 1500 fish features. The unsupervised classifier finds the most 5 representative feature classes for later identification.

E. Hadoop Ecosystem

The Hadoop ecosystem provides basis of distributed and parallel computing environment for machine learning algorithms. Over the Hadoop ecosystem, machine learning engines are accelerated through map-and-reduce mechanism and the Hadoop Distributed File System (HDFS) [6] stores large amount of historical fish photos (database image), texture patterns, pattern features, and environmental sensor data every time using the OFWare. These data along with feed and medication records allow development of fish breeding history. We adopt statistic tools, such as Mahout [7], to infer the relation among texture pattern, feed, medication, and breeding environment. This shortens the time required for heritage of breeding techniques since experiences of skilled staffs are reduced to a tutorial for apprentices. The inclusion of pricing, customers, and GPS data allows the development of business intelligence.

Shown in Fig. 5, the Hadoop Ecosystem was built based on Cloudera Distribution Including Apache Hadoop (CDH)[8]. Machine learning engines are designed using Spark or customized based on the CDH framework. Related CDH components including Apache Spark and their introductions are also available at [9] The CDH stack used by OFWare is as follows where Impala is primarily used for HBase access. Users' requests come from the Web are passed through the RESTful interface. Machine learning algorithms are implemented using Spark libraries and MapReduce APIs for taking the advantages of HDFS file system and clustered computing.

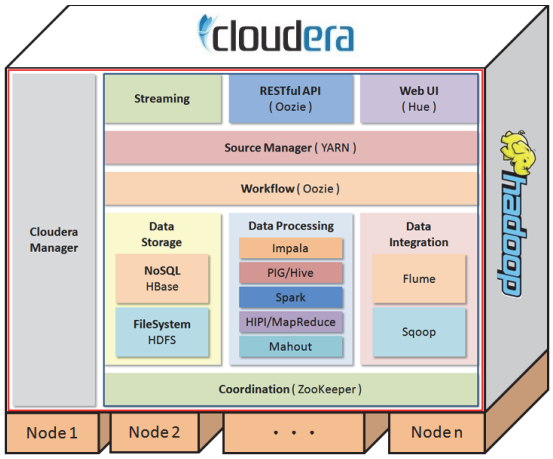

Fig. 5. The Hadoop Ecosystem for the OFWare system.

The Hadoop framework is for big data analytic purposes as introduced in the section Introduction. This paper primarily more focuses on the system level design providing a platform that continuously collects video and breeding data of ornamental fishes. With the platform and the Hadoop framework, the system can be easily scale up in the future. The execution efficiency of a larger system can be inferred according to the result given in the Abstract: identifying a fish feature among 500,000 koi fishes in 7 seconds if equipped with a small unit of a single name-node and data-node.

F. Machine Learning Engines

We build Spark machine learning modules [10] such as the support vector machine (SVM) [11] as well as customized machine learning modules such as the template matching in the Hadoop ecosystem. SVM predication allows the image retrievals by wrapping the newly extracted features as a database query. The OFWare updates the support vectors, which is the ML data model as illustrated in Fig. 1 by inclusion of the new features. Therefore, the OFWare records the latest features and representative close-up image each time when a staff member making an inventory. This overcomes the issue that shapes and texture patterns are time-variant as ornamental fishes grow. Moreover, in this way the OFWare detects newly purchased fishes by the fails of image retrievals. Then OFWare asks the staff user to create new profiles (with automatically generated IDs in Fig. 1) for the newly purchased fishes. Therefore, the OFWare is capable of incremental machine learning.

IV. System Operation and Results

We illustrate two input cases of the operation. The first case has a still input image of multiple koi fishes and the second one has an input video of another fish tank. Some users might just take photos of fishes for checking fishes' profile while some use videos to have dynamic postures of fishes.

A. Still Image Input

Fig. 6 and 7 illustrate the snapshots of operating the OFWare for making an inventory. Fig. 6(a) is the input image of a temporary tank with eight fishes. The OFWare allows human-assisted mark, the light-blue line, for separating overlapped fishes. Fig. 6(b) shows the database retrieval result of each fish. The upper row displays either the database fishes found or a question mark if the query fish in the lower row is not found in the database. The query failure is caused by either

a fish is new purchased or the close-up does not provide clear texture pattern of the fish. Clicking on a question mark or a database fish image, a user can browse database fish images that are the most similar to the query image and then select one for the inventory. Further clicking on one of the database image when browsing, the OFWare displays the profile details of the fish as show in Fig. 6(d).

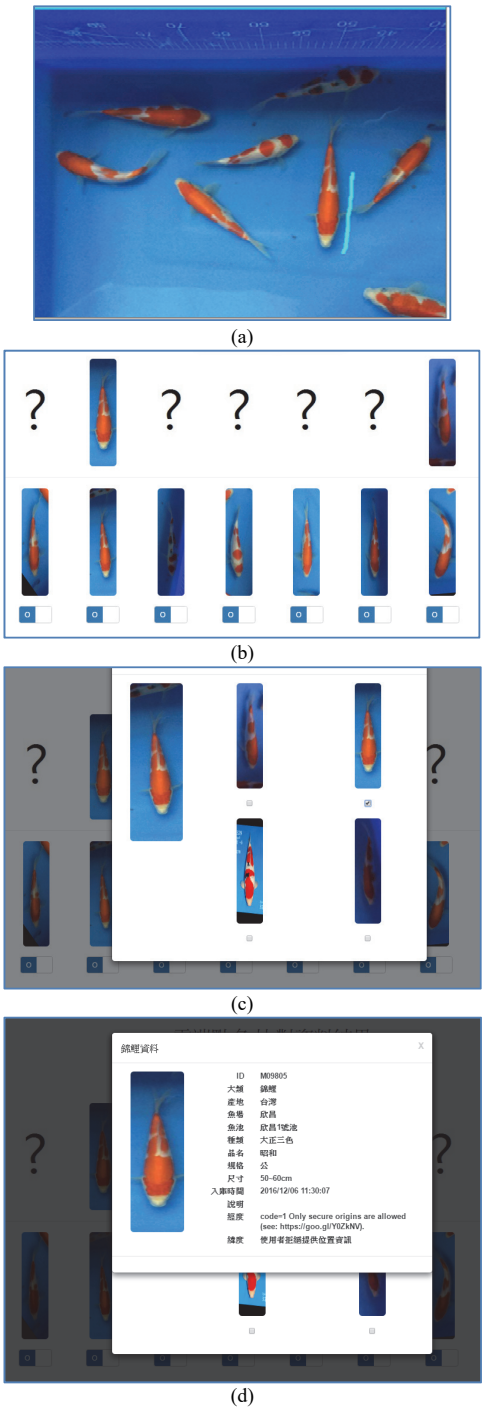

Fig. 6. The snapshot of system operation: (a) image of a temporary fish tank with eight fishes (seven of them are complete), (b) image retrieval result where two of the seven are found in the database (upper row: database images; lower row: query images), (c) the most similar images for manual check, (d) profile details of a fish.

B. *Video Input*

When a camera is set up above a fish tank for monitoring, the OFWare is able to make an automatic inventory. The operating procedure is already shown in Fig. 1. Fig. 7 (a)-(d) shows that occlusion, blurring, and glitter is obvious in the video. The OFWare conquers these interference by unsupervised classification and machine learning since a video provides more information for each fish than a single image. Fig. 7(e)-(g) shows the subsequent retrieval results and in them, images at the left are images taken at some old time instances while at the right are the latest images (representative query images).

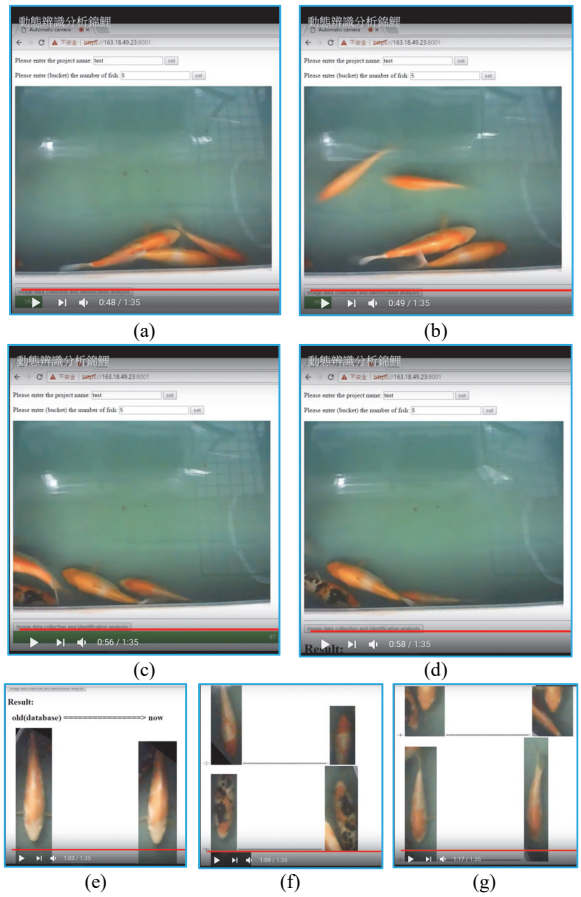

Fig. 7. The snapshots of system operation when using a video as input to the OFWare: (a)-(d)video clips, (e)-(g) subsequent image retrieval results.

V. Conclusion

In this paper we proposed a development platform, Ornamental Fish Warehousing (OFWare) system, for business intelligence, breeding techniques heritage and tutorial, certificates of owner, origin, competition, and related anti-forgeries based on the integration of information and communication technologies including big video data and mobile cloud computing. The platform is flexible such that customized and commercialized machine learning engines are easy to integrate into the system. Hadoop framework assures the scalability. System operation examples show the convenience and feasibility and achieve the automatic warehousing using video input. The proposed warehousing system becomes the basis for development of breeding history,

anti-forgery certificate, and aquaculture business intelligence in the future.

REFERENCES

[1] T. Bauernhansl, M. ten Hompel, and B. Vogel-Heuser, eds. *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien und Migration*. Wiesbaden: Springer Vieweg, 2014.

[2] Fielding, Roy T., and Richard N. Taylor. *Architectural styles and the design of network-based software architectures*. Doctoral dissertation: University of California, Irvine, 2000, ch. 5.

[3] Chao-Lieh Chen, Chia-Chun Chang, Chao-Chun Chen, et. al, "Developing an Ornamental Fish Warehousing System Based on Big Video Data," Intl. J. Automation and Smart Technology, North America, Issue 8, Jun. 2018. Available at: https://www.ausmt.org/index.php/AUSMT/article/view/1693.

[4] Tilkov, Stefan, and Steve Vinoski. "Node. js: Using JavaScript to build high-performance network programs," *IEEE Internet Computing* vol. 14, no.6, 2010, pp. 80-83.

[5] Tan, X. and Triggs, B., "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. *19*, no. 6, 2010, pp.1635-1650.

[6] Shvachko, K., Kuang, H., Radia, S., and Chansler, R. "The hadoop distributed file system," in *2010 IEEE 26th symposium on Mass storage systems and technologies (MSST),* May 2010, pp. 1-10.

[7] Alex Holmes, *Hadoop in practice*, Manning Publications Co., 2012.

[8] CDH tutorials, available at: https://www.cloudera.com/developers/get-started-with-hadoop-tutorial.html.

[9] CDH Components, Available at: https://www.cloudera.com/products/open-source/apache-hadoop/key-cdh-components.html.

[10] Meng, Xiangrui, et al. "Mllib: Machine learning in apache spark, "*Journal of Machine Learning Research,* vol.17, No.1, 2016, pp. 1235-1241.

[11] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers," *Neural processing letters vol.* 9, no. 3, 1999, pp. 293-300.