# An Experience Report on Building a Big Data Analytics Framework Using Cloudera CDH and RapidMiner Radoop with a Cluster of Commodity Computers

Sittiporn Kunnakorntammanop, Netiphong Thepwuttisathaphon, and Supphachai Thaicharoen(✉)

Faculty of Science, Srinakharinwirot University, Bangkok 10110, Thailand
{sittporn.kktn,netiphong.thep,supphachai}@g.swu.ac.th

**Abstract.** Many real-world data are not only large in volume but also heterogeneous and fast generated. This type of data, known as big data, typically cannot be analyzed by using traditional software tools and techniques. Although an open-source software project, Apache Hadoop, has been successfully developed and used for handling big data, its setup and configuration complexity including its requirement to learn other additional related tools have hindered non-technical researchers and educators from actually entering the area of big data analytics. To support big-data community, this paper describes procedures and experiences gained from building a big data analytics framework, and demonstrates its usage on a popular case study, Twitter sentiment analysis. The framework comprises a cluster of four commodity computers run by Cloudera CDH 6.0.1 and RapidMiner Studio 9.3 with Text Processing, Hive Connector, and Radoop extensions. According to the study results, setting up a big data analytics framework on a cluster of computers does not require advanced computer knowledge but needs meticulous system configurations to satisfy system installation and software integration requirements. Once all setup and configurations are correctly done, data analysis can be readily performed using visual workflow designers provided by RapidMiner. Finally, the framework is further evaluated on a large data set of 185 million records, "TalkingData AdTracking Fraud Detection" data set. The outcome is very satisfied and proves that the framework is easy to use and can practically be deployed for big data analytics.

**Keywords:** Apache Hadoop · Big data analytics · Cloudera CDH · Computer cluster · RapidMiner Radoop · Sentiment analysis

## 1 Introduction

Solving many real-world problems involves data that are not only large in volume but also heterogeneous and fast generated. The solutions for these problems cannot be practically implemented using traditional tools and techniques such as relational databases and typical in-memory data analysis tools. A number of commercial and


M. W. Berry et al. (Eds.): SCDS 2019, CCIS 1100, pp. 208–222, 2019.
https://doi.org/10.1007/978-981-15-0399-3_17

open-source software tools have been developed for coping with this type of data, and one of them that is stood out is Apache Hadoop.

Apache Hadoop is an open-source software library framework that has been successfully exploited by industries and academia when working with big data. With the framework, tasks of processing large data sets can be reliably and scalably distributed across a cluster of computers using a simple programming model called MapReduce. Apache Hadoop library framework consists of four main modules, Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop YARN, and Hadoop MapReduce. Although Apache Hadoop alone could be sufficiently deployed for handling large data sets, other software tools should also be concertedly applied in order to facilitate the implementation of the solutions for complex problems.

Hadoop ecosystem is a set of open-source software projects that has been productively employed together to provide software solutions for complex big data problems. It consists of Apache Hadoop and a number of related software components such as Pig, Hive, Spark, Oozie, Flume and Sqoop. Some examples of research studies and application domains that utilize Hadoop ecosystem are described as follows. Chennamsetty et al. used Hive as a data warehouse for storing large sets of Electronic Health Records (EHRs) and utilized HiveQL, a query processing language, to retrieve the stored data and generate reports for statistical analysis [3]. Sangeeta presented a study on using FLUME and Hive tools for twitter sentiment analysis [12]. In the study, FLUME was used to retrieve and collect tweets from Twitter, and HIVE Serde was operated to import tweets in JSON format into Hive table. Finally, some statistical analyses were performed through HiveQL query statements. Although these studies presented interesting use cases and applications of Hadoop ecosystem, their analyses were implemented on a single machine. To truly appreciate big data framework capability, analysis tasks should be carried out in a cluster of computers.

The following research studies are selected examples that implemented big data analysis on a cluster of computers. Liu et al. built a research data science platform using 40 industrial computers donated by Yahoo [9]. The purpose of their use of cluster computers rather than cloud-based architecture is to fully understand the Hadoop ecosystem and to use the platform for research studies. The platform is run by Apache Hadoop 2.7.2, HBase 1.1.5, OpenTSDB 2.2.0, and Spark 1.6.1, and evaluated on four test data sets. Logistic regression in Spark ML was selected to execute Wordcount MapReduce program. They found that size of the data sets, the number of CPU cores, and the driver memory contribute to performance. Durby et al. exploited Hadoop MapReduce ecosystem with multi-layered feed forward neural networks for stock market prediction [4]. In their study, a Hadoop cluster of 50 worker nodes run on Ubuntu 1.0.4 is utilized, and Neural Network was configured for parallelism using MapReduce. On the performance evaluation, they found that increasing number of nodes in the cluster can speed up the neural network training time. Even though building a big data analytics platform with Hadoop ecosystem allows implementers to fully understand the underlying principles of big data analytics, it requires installing a number of software tools and configuring them to work together. These complex installation and configuration could frustrate non-technical and inexperienced users on big data to advance into big data analytics domain. An alternative approach is to utilize an enterprise-ready open-source software suite.

Numerous software suites for manipulating big data are available in both commercial and open-source products. Different products have their own advantages and disadvantages. Therefore, which software to choose is dependent on criteria and preferences of the implementers. A research study by Nereu et al. evaluated five big data analytics platforms – Apache Hadoop, Cloudera, Spark, Hortonworks, and HPCC to find out which platform is best fit to small and medium-sized enterprises and non-government organizations [10]. Based on their study, they suggested that Cloudera is better than other platforms for all contexts, specifically when dealing with real-time large data sets. Ivanov et al. reported a performance evaluation on two enterprise big data platforms, DataStax Enterprise (DSE) and Cloudera's Distribution of Hadoop (CDH), using HiBench benchmark suite [7]. From their experimental results, CDH performs better than DSE in almost all test cases with faster execution time, faster read time, and higher throughput on Wordcount and enhanced DFSIO tests. For these reasons, Cloudera CDH is chosen in our study.

After a big data hardware and software platform architecture is established, analyzing data can be realized by either writing computer programs or using software tools. Although writing computer programs is more flexible and more powerful than using software tools, it is generally suitable to only computer and IT professionals. In many cases, simply using software tools can give analyzers enough functionality and analysis power for their tasks. For example, Han *et al.* used Weka for Chinese document clustering [6], Feltrin explored a usage of KNIME for geoscience application domain [5], and Tripathi *et al.* performed a sentiment analysis using RapidMiner [14]. Some comparative studies were conducted to evaluate these tools. Jovic *et al.* compared the following free software tools: RapidMiner, R, Weka, KNIME, Orange, and Scikit-learn, for general data mining projects [8]. They found that RapidMiner, R, Weka, and KNIME contain most of the desired characteristics for a fully-functional data mining platform. Another comparison was conducted by Altahi *et al.* on 19 open-source tools for data mining and knowledge discovery tasks. They found that Weka, Knime, and RapidMiner Studio are the most promising tools for their two evaluation criteria [1]. Since RapidMiner Studio offers visual workflow designer that greatly facilitates data analysis in addition to fully-functional data mining features similar to other tools, it is selected to be used in our study.

Finally, a number of quality criteria should be taken into consideration when building a big data analytics framework. Singh and Reddy presented an investigation of different big data analytics platforms to assess their strengths and weaknesses on the following six performance metrics – scalability, I/O performance, fault tolerance, real-time processing, data size support, and support for iterative tasks [13]. Using a 5-star rating, they provided an overview table summarizing performance scales of different platforms under the study. For example, Peer-to-Peer (TCP/IP) receives 5 stars for scalability and data size supported. Security is another important quality metric for big data analytics framework. Bhathal and Singh presented a study on different types of vulnerabilities of Hadoop framework and proposed some possible solutions to reduce the security risks [2]. They additionally implemented some security attacks to truely understand the security weaknesses of the Hadoop environment. From the results, they discovered that security modules provided by enterprise software suites such as IBM, MapR, Hortonworks, and Cloudera are not sufficient. Security breaches are still

possible. That is because securing Hadoop environment not only involves preventing unauthorized access to Hadoop and stored data but is concerned with network security and operating systems security as well. In our study, data size supported, scalability and ease of use are our main focus.

Based on our experiences and to the best of our knowledge, despite the fact that many current real-world data analysis problems are involved with big data, most works, particularly in academia, have been conducted on a small sample data set using a single machine and in-memory software tools. Consequently, the results of the analyses generally lack comprehensiveness and are useful only for ad-hoc studies. This may be because people inexperienced with big data might perceive that building a big data analytics platform requires expensive hardware and software and should be done only by highly-technical professionals. In this paper, we show that this belief is no longer true. We present a big data analytics framework using only a set of commodity computers, an enterprise-grade open-source big data software suite, Cloudera CDH, and an easy-to-use data science tool, RapidMininer Studio. In addition, we demonstrate the usability of the framework on Twitter sentiment analysis and prove the practicality of the framework on a large data set. The experimental result is very satisfactory.

The contribution of this paper is threefold. First of all, it presents detailed descriptions and suggestions on how to build a simple, accessible and affordable big data analytics framework that uses only commodity hardware and open-source software. Secondly, it provides a demonstrating example on how to use the framework to analyze a complex data analysis problem, Twitter sentiment analysis. Finally, it gives performance evaluation on a large data set to confirm the practicality of the framework.

The content in this paper is organized as follows. Section 2 gives a brief overview of technical background. Section 3 provides a description of procedures and methods. Section 4 presents the study results. Finally, Sect. 5 concludes the study.

## 2 Technical Background

This section briefly describes concepts and technologies for understanding the methodology in this paper. The content is divided into three parts: Cloudera CDH, RapidMiner Radoop, and sentiment analysis.

### 2.1 Cloudera's Distribution of Hadoop (CDH)

Cloudera's Distribution of Hadoop (CDH) is a 100% open source enterprise-grade big data analytics platform distribution provided by Cloudera. It integrates Apache Hadoop Core with a number of key open-source Apache projects such as Accumulo, Flume, HBase, Hive, Hue, Impala, Kafka, Pig, Sentry, Spark and Sqoop. Cloudera CDH allows enterprises to perform end-to-end big data workflows right out of the box. Moreover, it provides Cloudera Manager (CM), a Web-based management tool for managing CDH clusters, helping the installation process, and containing functionality for cluster configuration, resource allocation, and real-time monitoring. Cloudera also offers a virtual machine version, Cloudera CDH QuickStart, for interested users to be familiar with the platform. Users can install it in a single machine for their own

exploration and experimentation. Cloudera CDH Quickstart is available in three favors, VirtualBox, VMWare and Docker.

In this study, to really experience a real-world big data analytics platform as possible, rather than using Cloudera QuickStart virtual machine, a complete CDH distribution was installed in a cluster of commodity computers. Moreover, instead of utilizing the latest CDH 6.2.0, CDH 6.0.1 was chosen for our framework because all of its software component versions are supported by the current version of RapidMiner Radoop 9.3. Unsupported versions of some CDH components can cause connection problems between CDH and RapidMiner Radoop. Cloudera CDH 6.0.1 is available at https://www.cloudera.com/downloads/cdh/6-0-1.html.

### 2.2 RapidMiner Radoop

Radoop is a plug-in extension of RapidMiner Studio introduced by Prekopcsak *et al.* [11]. It provides code-free operators for using Hadoop, Hive, and Spark to carry out a number of data analysis and data mining tasks such as Naive Bayes, Linear and Logistic Regressions, Decision Tree, Random Forest, Support Vector Machine, K-Means Clustering, Scoring, and Validation. In addition, it offers a visual workflow designer and data processing operators similar to typical in-memory RapidMiner operators but can be run in-parallel in Hadoop clusters. Such data processing operators are data access, data blending and data cleansing. **Hadoop Nest** operator is the main operator in Radoop. To run processes synchronously in a Hadoop cluster, all operators must be executed inside **Radoop Nest** operator.

In this paper, since Radoop does not yet provide operators for text processing, text pre-processing tasks were carried out in a local computer using typical in-memory RapidMiner Studio operators. However, predictive modeling and evaluation tasks for sentiment analysis were performed in a Hadoop cluster within **Hadoop Nest** operator.

### 2.3 Sentiment Analysis

Sentiment analysis is an automatic process for determining the sentiment polarities of people opinions whether they are positive, negative, or neutral. These opinions usually are presented in a textual format. Sentiment analysis has been applied to many application domains such as product/movie review, customer services, crime mitigation, and stock prediction.

Sentiment analysis approach can be divided into three main categories: (i) lexical-based method, (ii) supervised-learning method, and (iii) hybrid method. Lexical-based sentiment analysis relies on extracting and mapping words to a sentiment category (positive, negative, or neutral) and then uses the mapping results to compute sentiment scores. Subsequently, a threshold is applied to the total score calculated from all words in the sentence to determine the polarity. In contrast, supervised-learning method learns a classification model from labeled data of text representation and utilizes the model to predict the polarity of the whole textual sentence. Finally, the hybrid method exploits both lexical and supervised-learning approaches.

In this paper, a machine-learning approach, Logistic Regression, was used for predicting sentiments of Twitter data.

## 3 Methodology

In this section, detailed descriptions of setting and configuring a Cloudera CDH cluster, connecting Cloudera CDH to RapidMiner Radoop, and Twitter sentiment analysis are presented.

### 3.1 Build a Cloudera Cluster

In this study, a cluster of four computers is constructed. It consists of one TP-Link Gigabit router and four desktop computers. The router was configured to assign fixed private IP addresses to these computers as 192.168.0.2, 192.168.0.3, 192.168.0.4, and 192.168.0.5, respectively. CentOS 7 was then installed in all computers.

Due to compatibility requirements, rather than installing the latest Cloudera CDH 6.2.0, a previous version, Cloudera CDH 6.0.1, was selected. The reason is that the highest version of Spark supported by RapidMiner Radoop 9.3 is Spark 2.2 which is available in Cloudera CDH 6.0.1. Based on our experiences, version incompatibility can cause a connection problem between Cloudera CDH and RapidMiner Radoop later during the data analysis process.

A computer with 16 GB of RAM was chosen as a name node and the remaining three computers with 8 GB, 8 GB, and 12 GB of RAM as data nodes. It is recommended to use a computer with the largest size of RAM to act as a name node because it generally performs many roles. Based on our experiences, with similar CPU speed and number of cores, the larger the RAM size, the smoother an analysis process is running.

Before proceeding to install Cloudera CDH, it is highly necessary to configure the system of all computers in the cluster to meet all the requirements. Some examples of Linux commands used in our study are given below. Note that lines preceded by a ">" are corresponding to Linux commands, lines preceded by "#" are comments, and those lines without ">" or "#" are content of a file open from their preceding command.

- Setting up time zone
  *>timedatectl list-timezones | grep Asia*
  *>timedatectl set-timezone Asia/Bangkok*
- Configuring and synchronizing NTP services
  *>sudo yum install ntp*
  *>sudo nano/etc./ntp.conf*
  *### Comment out existing public servers such as server 0.centos.pool.ntp.org burst*
  *### and add the following three servers instead.*
  *### The server names suitable to a specific country in Asia can be found at*
  *### https://www.pool.ntp.org/zone/@*
  *server 1.th.pool.ntp.org iburst*
  *server 3.asia.pool.ntp.org iburst*
  *server 1.asia.pool.ntp.org iburst*
  *>sudo systemctl start ntpd*
  *>sudo systemctl enable ntpd*

*>sudo ntpdate-u 1.th.pool.ntp.org*
*>sudo hwclock–systohc*

- Disabling SELinux
  *>sudo nano/etc./selinux/config*
  *### Change the line: SELINUX = enforcing to SELINUX = permissive or*
  *### SELINUX = disabled*
- Disabling Firewall
  *>sudo systemctl disable firewalld*
  *>sudo systemctl stop firewalld*
- Disabling IPv6
  *>sudo nano/etc./sysctl.conf*
  *net.ipv6.conf.all.disable_ipv6 = 1*
  *net.ipv6.conf.default.disable_ipv6 = 1*
- Configuring host names (change the host name to match names of individual computers in the cluster)
  *>sudo nano/etc./sysconfig/network*
  *HOSTNAME = master.xxx.xx.xx*
  *>sudo nano/etc./hosts*
  *192.168.0.2master.xxx.xx.xxmaster*
  *192.168.0.3node1.xxx.xx.xx node1*
  *192.168.0.4node2.xxx.xx.xx node2*
  *192.168.0.5node3.xxx.xx.xx node3*
  *127.0.0.1 localhost*
  *::1*

The first step for installing Cloudera CDH is to download and install Cloudera Manager in a computer established as a name node (or master host), which can be done by the following three commands.

*>wget https://archive.cloudera.com/cm6/6.0.1/cloudera-manager-installer.bin*
*>chmod u + x cloudera-manager-installer.bin*
*>sudo./cloudera-manager-installer.bin*

With Cloudera Manager successfully installed, the next step is to log into Cloudera Manager and install Cloudera CDH. This step can be carried out as follows: (i) open a Web browser, (ii) point to Cloudera Manager log-in page such as http://yourhostname:7180/cmf/login, and (iii) log into the system. The default username and password are admin/admin. After logging in, the installation process begins. Based on our available hardware resources, Cloudera Express version was adopted, and according to the scope of our data analysis tasks, Data Engineering services (consisting of HDFS, YARN, ZooKeeper, Oozie, Spark, Hive, and Hue) were installed. During the installation process, Cloudera CDH services can also be distributed to be installed in computer data nodes by specifying their host names. Some memory configurations such as VM Swappiness and memory allocation may need to be performed during and after the installation for performance optimization and for fixing some warnings.

The final step for building a Cloudera cluster is to install RapidMiner Studio and three extensions – Text Processing, Hive Connector, and Radoop, in the computer name node.

### 3.2 Connect Cloudera CDH with RapidMiner Radoop

After completing a Cloudera CDH cluster setup, a connection between Cluster CDH and RapidMiner Radoop can be established by using "Import from Cluster Manager" option under "Manage Radoop Connections" menu of RapidMiner Studio. With this approach, almost all parameters and values are already configured except Spark version. In this paper, Spark version 2.2 was selected which is the highest version supported by the latest version RapidMiner Radoop 9.3 at the time of writing.

Finally, to be certain that all software services can be effortlessly working together, a series of tests should be performed on individual Radoop settings and ended with a complete full/integration test.

### 3.3 Analyze Tweet Sentiments on the Topic of Global Warming and Climate Change

In this study, sentiment analysis problem was chosen for demonstrating the usability of the proposed framework because it is more complex than a typical classification problem. Sentiment analysis is a text classification problem whose tasks can be divided into two steps: (i) text pre-processing and (ii) predictive modeling and evaluation. Text pre-processing is a process of cleansing and transforming raw text into a representation that is suitable to further analysis such as an input to a machine-learning algorithm. Two commonly known text representation formats to date are vector space model and word embedding where the first format was used in this paper. Predictive modeling and evaluation is a process of building a predictive model from one set of data and evaluating the model performance using another data set. The first data set is called a training set and the later is a test set.

Since the current version of RapidMiner Radoop does not yet provide operators for handling text data, text pre-processing tasks were carried out locally in computer name node. All other tasks, predictive modeling and evaluation, were implemented and executed concurrently in the Hadoop cluster.

**Data.** Two data sets were used in this study.

- The first data set was downloaded from https://www.figure-eight.com/data-for-everyone/, which is a publicly-available sentiment analysis data set on the topic of global warming and climate change. This data set contains 5,679 tweets with three types of polarity: Yes, No, and N/A ('Yes' means that the owner of a tweet believes that global warming and climate change are really happening, 'No' means the opposite, and 'N/A' is corresponding to a missing label). This data set was used for demonstrating processes of building a predictive model and evaluating the model performance using RapidMiner Radoop operators on textual data.
- The second data set was directly retrieved from Twitter using **Search Twitter** operator in RapidMiner Studio. The search keyword is "Global Warming and

Climate Change-filter:retweets AND-filter:replies", where filters are added to the keyword search to eliminate duplicate tweets. This data set was used for illustrating how tweets can be retrieved and how a new text representation data set can be constructed using a set of features/words defined from another data set.

**Text Pre-processing.** Text pre-processing process is divided into two parts. The first part is building a training set and the second part is building a test set. As described previously, these two parts are run locally in the computer name node using typical in-memory operators of RapidMiner Studio.

As shown in Fig. 1, started from the top-left operator, a twitter data set in CSV format is read into RapidMiner workspace in a tabular format, followed by selecting only relevant attributes, filtering out undesired rows of data, setting one column as a class label, and sampling a subset of data (500 rows of "Y" and 500 rows of "N"). When text data are imported into RapidMiner, they are automatically converted to nominal data. As a result, before feeding them into **Process Document from Data** operator, they need to be converted back to text.

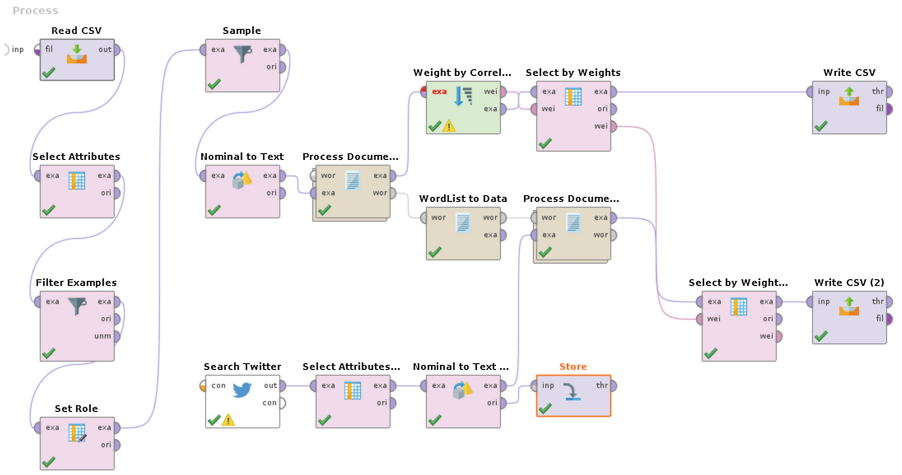


**Fig. 1.** An overview of text processing visual flow diagram.

Inside **Process Document from Data** operator, a series of text pre-processing tasks was performed as follows: tokenization, case transformation, English stopword removal, user-defined stopword removal, and word stemming (shown in Fig. 2). The outputs from this operator are a vector space model with TF*IDF weighting scheme and a list of words as a dictionary. A statistical correlation was used for weighting importance of individual words and the top 150 correlation words were selected as features.

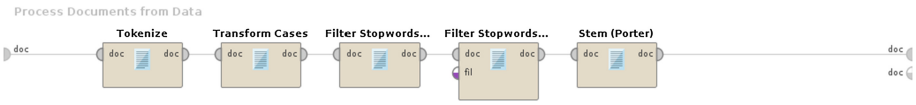


**Fig. 2.** Text processing visual flow diagram for the tweet data set.

Figure 3 shows a sequence of text pre-processing tasks inside another **Process Document from Data** operator for the retrieved tweets.

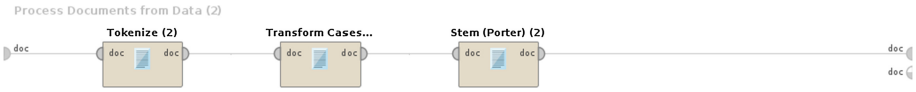


**Fig. 3.** Text processing visual flow diagram for the retrieved tweets.

Text representations produced from the downloaded data set and retrieved data set were saved into two CSV files for the next step (see **Write CSV** operators in **Fig.** 1).

**Predictive Modeling and Evaluation.** Predictive modeling and performance evaluation processes are illustrated in Figs. 4, 5 and 6. Since some data files are saved into a local computer, these processes contain a combination of some in-memory operators and in-cluster operators. Note that all in-cluster operators must be put or executed inside **Radoop Nest** operator.

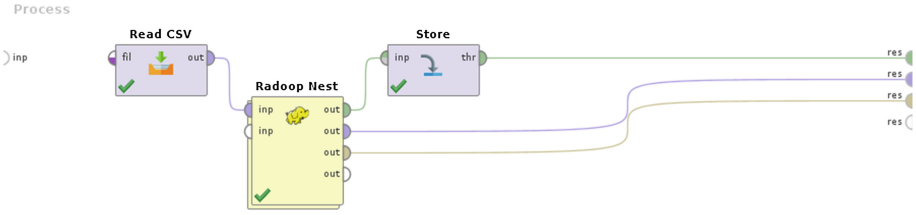


**Fig. 4.** An overview of building and evaluating a predictive model in Hadoop cluster.

In Fig. 4, a text representation data set is read from a local data file created in the previous step and fed into **Radoop Nest** operator. The output from **Radoop Nest** operator is a predictive model which is saved into a data file and will be used for making predictions on the retrieved tweet data set.

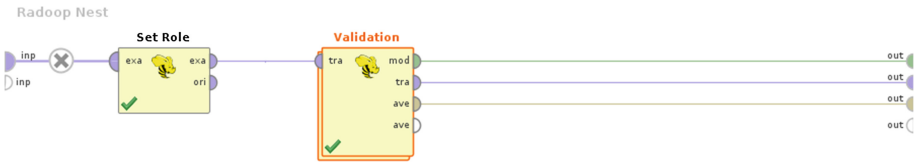


**Fig. 5.** Inside Hadoop Nest operator, a training data set is split for building evaluating a predictive model using Split Validation operator.

In Fig. 5, a sentiment column is set as a class label and the data set is split into training set and test set using **Split Validation** or **Validation** operator. Its three outputs are a predictive model, training data set (optional), and the classification performance.

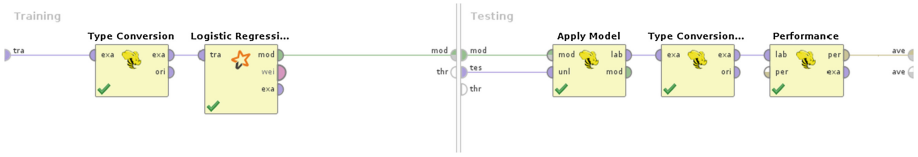


**Fig. 6.** Inside Split Validation operator, a logistic regression model is built and evaluated.

Figure 6 shows a sequence of operators for building and evaluating a predictive model inside **Split Validation** operator.

After having a predictive model, the model can then be used to predict the polarities of tweets retrieved from Twitter as shown in Figs. 7 and 8. The **Retrieve** operator retrieves a predictive model saved in RapidMiner workspace and **Read CSV** operator reads text representation data set of the retrieved tweets created in the text pre-processing step. **Apply Model** operator applies the predictive model to predict polarities of the retrieved tweets. In this case, the prediction results must be manually verified by the researchers.


**Fig. 7.** Predictive model and pre-processed tweets are fed into **Radoop Nest** operator for predictions.


**Fig. 8.** An **Apply Model** operator is used for making predictions on the retrieved tweets.

### 3.4 Evaluate the Framework on a Large Data Set

To evaluate the practicality of the framework for real-world big data analysis, an additional experiment was conducted on a large data set, "TalkingData AdTracking Fraud Detection". The data set (training.csv), was downloaded from Kaggle. It consists of eight attributes, 185 million rows of data with the size of 7.54 GB. For this size of data, processing it using traditional software tools on a typical computer with 16 GB of RAM can cause the system to freeze. For the presented framework, the data set was used for building a predictive model and evaluated the model performance successfully. The whole process took approximately one and a half hour to complete. Figures 9 and 10 display the underlying processes for building and evaluating a predictive model using a large data set.

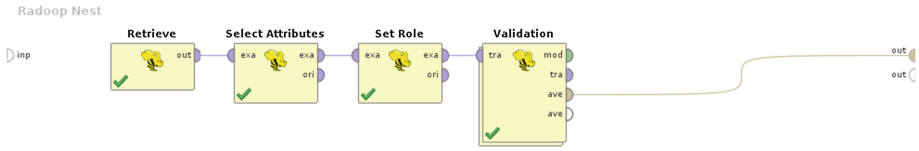


**Fig. 9.** Retrieve a large data set previously stored in Hive and feed it into Radoop Nest.

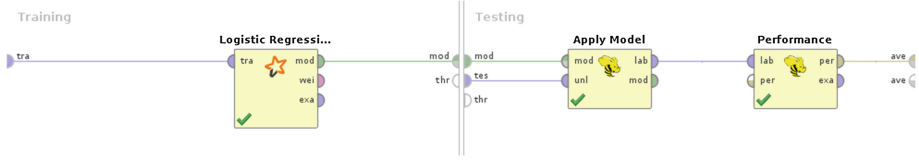


**Fig. 10.** Build a predictive model using Logistic regression and evaluate its performance.

## 4 Experimental Results

A Cloudera cluster with its hardware and software specifications resulted from our study is illustrated by a UML deployment diagram in Fig. 11.

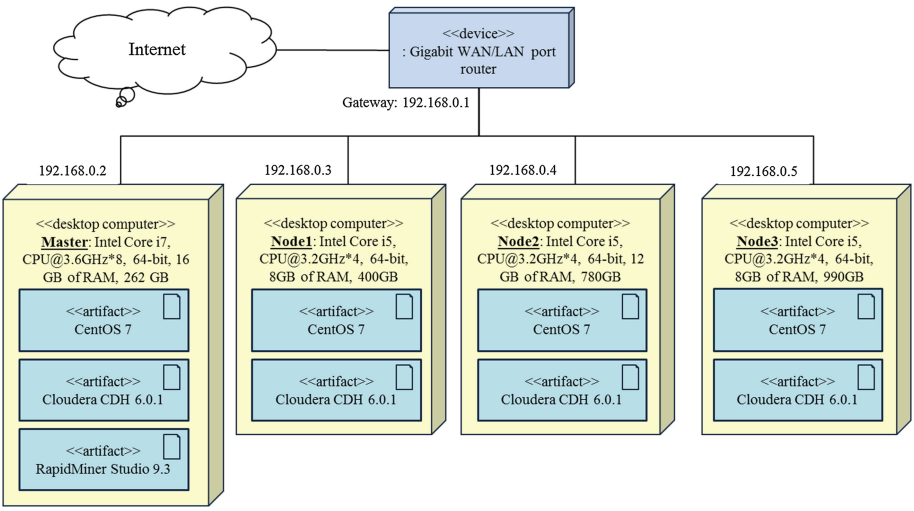


**Fig. 11.** A cluster of four commodity computers with hardware and software specifications where the master computer acts as a name node and node1, node2 and node3 computers function as data nodes.

As shown in Fig. 11, the cluster consists of one master host and three worker hosts. The master host is set as a name node and worker hosts as data nodes. Since RapidMiner Studio is installed in the master host, all in-memery data analysis tasks will be performed on this computer. Therefore, the master host is set to have the largest RAM size among all computers in the cluster.

Data in Table 1 is the Twitter data analysis results generated by RapidMiner Studio as a classification performance matrix in terms of accuracy, precision, and recall. It is given for only confirming the usability of the framework.

**Table 1.** Classification performance.

| | True Y | True N | Class precision |
|---|---|---|---|
| Pred. Y | 131 | 22 | 85.62% |
| Pred. N | 24 | 125 | 83.89% |
| Class recall | 84.52% | 85.03% | |

(Accuracy = 84.77%)

## 5 Conclusions

In this paper, a big data analytics framework was constructed, a case study on twitter sentiment analysis was presented, and an experiment of data analysis using a large data set was performed. The framework is built on a cluster of four commodity computers in which one computer is set as a name node and the remaining ones are set as data nodes.

Cloudera CDH 6.0.1 and RapidMiner Studio 9.3 with Text Processing, Hive Connector, and Radoop extensions are installed on the name node computer.

Based on the study results, a big data analytics framework can be constructed by using only commodity computers and open-source software applications. Building a successful framework requires careful software configurations and detailed examination of supporting software versions between Cloudera CDH and RapidMiner Radoop. In our study, instead of selecting the latest version, an earlier version, Cloudera CDH 6.0.1, is used because it contains Spark 2.2, the highest version supported by RapidMiner Radoop at the time of writing. In addition, based on our experiences in this study, size of RAM is one of key factors for efficiently conducting a data analysis. As a consequence, it is very important to check Cloudera hardware requirements at an early stage of a project. For the name node computer, we recommend using RAM whose size is larger than that specified in the Cloudera requirements because it is also used for running RapidMiner Studio/Radoop processes.

Cloudera CDH software suite allows Hadoop ecosystem to be readily set up on a cluster computers. Visual workflow designer in RapidMiner Studio makes big data analysis tasks become easier for people with no programming experiences. Therefore, we hope that the methodology and framework presented in this paper can be used both as a starting point of learning for researchers, educators, or professionals in any domain who are interested in the area of big data analysis and as a research and big data analysis tools for experienced users.

For future work, the framework could be used to collaborate with sensor devices from Internet of Things system for real-time big data analysis.

## References

1. Altalhi, A.H., Luna, J.M., Vallejo, M.A., Ventura, S.: Evaluation and comparison of open source software suites for data mining and knowledge discovery: open source software suites for data mining and knowledge discovery. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **7**(3), e1204 (2017)
2. Bhathal, G.S., Singh, A.: Big data: Hadoop framework vulnerabilities, security issues and attacks. Array **1**, 100002 (2019)
3. Chennamsetty, H., Chalasani, S., Riley, D.: Predictive analytics on electronic health records (EHRs) using Hadoop and Hive. In: 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1–5. IEEE, Coimbatore (March 2015)
4. Dubey, A.K., Jain, V., Mittal, A.P.: Stock market prediction using Hadoop MapReduce ecosystem, p. 6 (2015)
5. Feltrin, L.: KNIME an open source solution for predictive analytics in the geosciences [software and data sets]. IEEE Geosci. Remote Sens. Mag. **3**(4), 28–38 (2015)
6. Han, P., Wang, D.B., Zhao, Q.G.: The research on Chinese document clustering based on WEKA. In: 2011 International Conference on Machine Learning and Cybernetics, pp. 1953–1957. IEEE, Guilin (July 2011)
7. Ivanov, T., Niemann, R., Izberovic, S., Rosselli, M., Tolle, K., Zicari, R.V.: Performance evaluation of enterprise big data platforms with HiBench. In: 2015 IEEE Truscom/BigDataSE/ISPA, pp. 120–127. IEEE, Helsinki (August 2015)

8. Jovic, A., Brkic, K., Bogunovic, N.: An overview of free software tools for general data mining. In: 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1112–1117. IEEE, Opatija (May 2014)
9. Liu, F.C., Shen, F., Chau, D.H., Bright, N., Belgin, M.: Building a research data science platform from industrial machines. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 2270–2275. IEEE, Washington DC (December 2016)
10. Nereu, J., Almeida, A., Bernardino, J.: Big data analytics: a preliminary study of open source platforms. In: Proceedings of the 12th International Conference on Software Technologies, pp. 435–440. SCITEPRESS - Science and Technology Publications, Madrid (2017)
11. Prekopcsák, Z., Makrai, G., Henk, T., Gáspár-Papanek, C.: Radoop: analyzing big data with RapidMiner and Hadoop. In: RCOMM 2011: RapidMiner Community Meeting and Conference, p. 13. Rapid-I (June 2011)
12. Sangeeta: Twitter data analysis using FLUME & HIVE on Hadoop framework. Spec. Issue Int. J. Recent Adv. Eng. Technol. **4**(2), 119–123 (2016)
13. Singh, D., Reddy, C.K.: A survey on platforms for big data analytics. J. Big Data **2**(1), 8 (2015)
14. Tripathi, P., Vishwakarma, S.K., Lala, A.: Sentiment analysis of english tweets using rapid miner. In: 2015 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 668–672. IEEE, Jabalpur (December 2015)