ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BEHAVIOR CLUSTERING SYSTEM BASED ON LOCATION DATA FOR LABORATORY SAFETY MANAGEMENT

[1]Hyun-Seong Lee, [2]Seong-hyun Lee, [3]Jae-gwang Lee, and [4]Jae-kwang Lee

[1,3,4] Department of Computer Engineering, Hannam University, 34430, KOREA

[2] Information Security Research Division, ETRI, 34129, KOREA

E-mail: [1,3]{hslee,jglee}@netwk.hannam.ac.kr, [2] duribun2@gmail.com, [4]jklee@hnu.kr

ABSTRACT

The laboratory safety management system that can predict the risk situation and monitor the safety status. In order to predict and inform the researchers about the risk situation of the laboratory, it is necessary to classify the location area where the risk factor exists and the status information of the researcher according to the real time position. Based on the classification results of the location history data for the previous risk situation, classification algorithms such as K-Means or density-based spatial clustering of applications with noise (DBSCAN) are used to classify the real-time location. However, since the classification algorithm requires a large amount of computation, there is a problem that a high-grade processor must be used in order to process many position record data. To solve this problem, we use Apache Spark, which has recently become a big data processing framework. Since Apache Spark processes in memory and is suitable for iterative operation of large-scale data, it can perform classification operation of large amount of position data more quickly. In addition, Apache Spark supports RDD-based Matrix storage method to process location data type, enabling faster location data processing. In this paper, we design and implement a classification algorithm for location data stored in the Apache Spark environment. The classification algorithm uses the existing K-means algorithm and the DBSCAN algorithm more suitable for position data. Based on the classified result data, the classification speed of position data is compared and analyzed.

Keywords: *Laboratory Safety, Apache Spark, Big Data, Clustering Algorithm, DBSCAN*

1. INTRODUCTION

Research and development activities are actively being carried out in enterprises, universities, and research institutes. As the number of researchers increased, several countries developed safety management measures for safe research activities. In most research institutes and university laboratories, there is a growing interest in safety management, including safety training and the deployment of safety managers. However, laboratory accidents are occurring steadily.

According to the accident rate data of the Berkeley University Laboratory, accidents occurred at a steady rate since 2004 [1]. Various studies have been attempted to prevent such laboratory accidents. For example, information analytics research has been conducted to help researchers predict risk and take precautions based on a variety of information[2][3]. Recently, as a result of the development of technology, a system for monitoring information such as fire and gas leakage through a variety of sensors and wireless networks has been proposed[4]. However, since the existing laboratory safety management research is based on static data analysis, it cannot predict and prevent real-time accidents. In addition, research to predict accident risk by analyzing real-time monitoring data is insufficient.

In this paper, we propose a classification model for the area where accident risk exists as the location record data of researchers in the laboratory for laboratory safety management. The risk status can be predicted in real time by comparing the researcher's real-time location with the risk factor zone model. At this time, in order to classify by position, a lot of location record data should be classified into a clustering algorithm. Clustering algorithms are typically K-means and DBSCAN. However, clustering algorithms require many operations. Recently, various frameworks for processing such Big Data have been developed[5]. Among them, Hadoop is a framework used in various fields[6]. However, Hadoop makes a lot of I / O going through the disk. This may cause performance degradation when processing big data. Apache Spark can perform operations on memory or disk[7][8]. By supporting the batch process, it is possible to perform repetitive analysis, so that a lot of data processing for the same job can be performed more quickly. Also, Relational Database, which is used as an existing database, supports ACID (Atomicity, Consistency, Isolation as well as Durability), which makes it difficult to extend and apply complex transaction processing to big data. NoSQL is a research that supports big data and is suitable for distributed processing [9]. NoSQL simplifies data modeling and makes distributed storage and processing easier.

Therefore, the proposed system uses Apache Spark for clustering operation. In Apache Spark, the user's location record data stored in MongoDB is retrieved and transformed into RDD form, and the model is created by implementing the clustering algorithm K-Means and DBSCAN. Based on the generated model, it is possible to classify the researcher 's real-time location. Compares the computation speed with the processing time in the existing environment, and compares the classification speed. We also compare and analyze clustering algorithms.

2. RELATED STUDY

For safety management of the laboratory, a paper on descriptive analysis, multiple bar charts, and trend analysis based on accident data from 20 laboratories in Cross River State, Nigeria is presented in the paper "Evaluation of Evaluations of the Effect of Workshop / Laboratory Accidents and Precautionary Steps towards Safety Practice" Analysis of variance-SPSS) [3]. As a result of the analysis, it was reflected in the precautionary education. In addition, we proposed a human resource management method based on personal data such as researchers' characteristics, behaviors, preferences, learning abilities, and test scores in the article "Research on the Accident Prediction for Smart Laboratory Based on Statistical Analysis" [4]. Personal data is classified into risk groups and potential groups by the K-Means classification algorithm. Predict security incidents based on user learning contents and test results. However, this reflects the results of analysis based on static data in safety education, so that the cause of the accident can be prevented only indirectly.

In this paper, we propose a wireless security monitoring system in a chemical laboratory environment. The wireless sensor network node in the laboratory receives information and provides remote monitoring and control functions. Fire, and toxic gas leaks are detected, we propose a system that sends alerts to researchers in case of danger through an estimation algorithm. The monitoring system can quickly evacuate to an emergency and maximize the safety of the researcher. However, it notifies after the accident that it reduces accident damage, but ultimately it does not prevent accident occurrence. In this way, the existing laboratory safety management system was able to reflect the accident data analysis results in education or to prevent the sensor data from being monitored in real time. However, research on accident prediction service based on real-time information of researchers is insufficient.

2.1 Big Data Analysis Cluster Platform

With the advent of the concept of Big Data, the demand for distributed computing for large-scale data analysis has increased dramatically. In particular, Apache Hadoop's MapReduce framework is widely used for big data analysis. Hadoop provides a MapReduce engine that allows you to easily perform parallel computations without worrying about big data storage, distributed engines, and complex structures or defects. These OS-level abstractions have been used by many people. However, in this framework, every time the process is repeated, the data is directly stored on the disk, so the I / O is increased and the performance is deteriorated. In the case of a clustering algorithm in which the same processing operation is frequently repeated, each repetition can be created by a single MapReduce operation, but performance is degraded because there is a process of reading and processing the same data on the disk in each iteration procedure. Apache Spark supports in-memory processing and solves this problem by using a robust Resilient Distributed Datasets (RDD) storage structure for distributed processing.

2.1.1 Apache Spark

Apache Spark is a general purpose distributed high performance cluster platform. Spark provides a system structure and tools that can process data quickly, such as in Hadoop, when iterative data operations are required in bulk data processing. It enables operations on distributed nodes and can perform memory-based operations. I / O speed is fast because the operation is performed in memory. You can also use the directional noncircular graph (DAG) engine to generate efficient queries for data transformation.

Spark consists of one master node and several worker nodes in a cluster. When the master node receives the task, it loads the corresponding data in RDD format. A dataset constructed in the RDD format consists of partitions divided into several nodes. Partitioned partitions are stored in each worker node memory and operations are performed. Figure 1 shows the Spark system architecture.

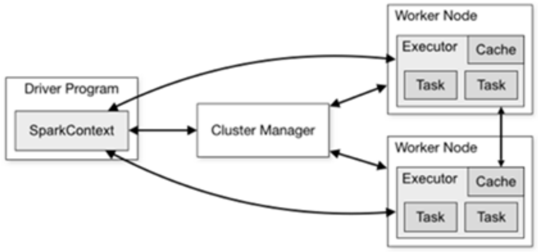


Figure 1: Apache Spark System Architecture

Spark is written in Scala language can manipulate data interactively through a scalar shell. Scala is compiled into the Java Virtual Machine (JVM) and converted to Java classes. This works with Hadoop code and easily track because it is convenient to convert Java code. It is also possible to apply it faster to use existing Java libraries.

2.1.2 *RDD (Resilient Distributed Datasets)*

RDD is the core programming abstraction of Apache Spark's core programming abstraction. It has the property that it can be read only in the form of distributed shared memory without changing value. To do the work in Spark, we convert the data to RDD and transform the RDD into various forms to perform the operation. Spark automatically splits the RDD data into several pieces and stores it in a cluster. RDD is performed by delay calculation, and when it is created, it does not load the data but loads the data only when the calculation is necessary. This enables fast processing because it minimizes unnecessary I / O and optimizes repeated operations.

There are two types of RDD operations: transformation and action. Transformation is an operation that takes an existing RDD as input and transforms it into a new RDD. Delay calculations are applied by filter and map operations and data is not loaded until action is performed. Action performs an operation and saves it as a file or creates result information. For example, the Action operation collect is the task of writing the result to memory, and all previous transformation operations are performed together.

2.2 NoSQL Database

NoSQL is a distributed database technology. Unlike relational database (RDB), it is composed of non-relational data repositories [11]. Master-slave type NoSQL stores data in a large number of slave nodes and stores and manages all meta information in one master node. It is relatively easy to add or delete servers in the cluster. In addition, it is possible to construct an hexa-byte database in Tera, which is difficult to process in an existing RDB by distributing key-value format data like Hash Map. Since the storage structure is not complicated, a large amount of data I / O can achieve excellent performance without bottleneck.

2.3 Clustering Analysis of Accidents

Cluster algorithms classify a given number of data into groups of similar characteristics. This is a kind of unsupervised learning, and no target value or label is given for the input value. It is an algorithm that classifies clustered data from a large number of data, such as the original un-clustered data of Figure 2. In particular, K-Means and DBSCAN are widely used among clustering algorithms. In this paper, we propose a clustering algorithm based on the location data of the GPS trajectories.

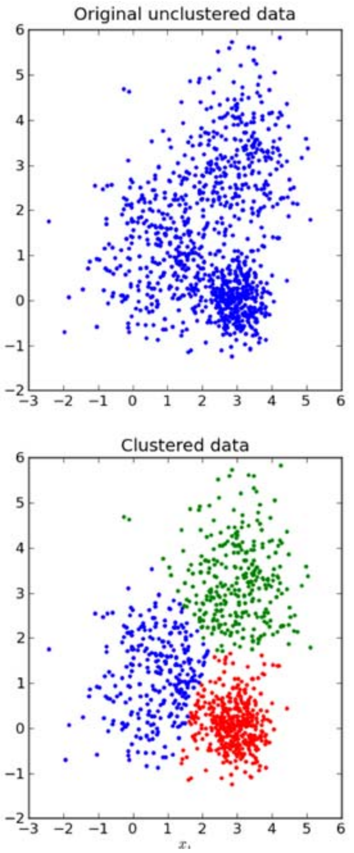


Figure 2: Example of Clustering Algorithm

K-Means categorizes data closer to each other. In this case, there is one center point for each group, and the cost is defined as how close each data is to its center. K-Means reduces this cost as much as possible according to the formula below.

$$\min_{b,w} \sum_i^n \sum_j^k w_{ik} \parallel x_i - b_j \parallel _2^2 s.t.\ \sum_j w_{ij} = 1, \forall j \quad (1)$$

Assume that there are n data and k clusters. binary variable is. In this case, b_j denotes the center of the j-th cluster and w_{ij} is a binary variable having 1 if the i-th data belongs to the j-th cluster or 0 otherwise. The following condition is a constraint that each piece of data must belong to one cluster. This formula can be found to be optimal by verifying all group combinations. However, if we fix b or w, we can calculate the mean of the j-th group. It is the k-means algorithm to find these 'k' mean. The K-means does not converge to the global optimum but converges to the local optimum and is vulnerable to initial value setting. Also, the K value must be entered, and the classification result is clearly different according to the K value.

In the paper, "K-Means Clustering Algorithm to Examine Patterns of Vehicle Crashes in Before-After Analysis," the K-Means algorithm was used to analyze the pattern of vehicle crashes. All similarities were counted and clustering was grouped together. For each group, we found an average value of various characteristics (geometry, environment, accident, etc.). The risk index for the group was also set to allow the risk level to be set. Using the analyzed information, an accident prediction model was developed and applied to decision support.

The DBSCAN algorithm takes the maximum radius and the minimum number of groups as inputs and classifies them into groups and noises through a four-step procedure. First, the distance of the remaining points is calculated at an arbitrary point of the minimum number of groups. And calculates the number of points within the maximum radius of the values. Second, if the number of points in the radius is greater than the minimum number of groups given, add the group with the point as the center point. Third, the same method is applied at the points included in the minimum radius at the center point to expand the cluster if the second condition is satisfied. Through the expansion of these clusters, sections longer than the reference length can be processed into a single cluster. Finally, the points that are not included in any cluster are treated as noise. Unlike the K-Means algorithm, it is used when it is difficult to specify the number of groups because there is no need to set the number of groups. In addition, the concept of noise exists, which can reduce errors and errors due to unnecessary data.

In the paper, "Comparison of clustering techniques for traffic accident detection" using DBSCAN algorithm, we compared DBSCAN and K-Means algorithms for vehicle speed and position in simulated traffic accident simulations. As a result, the K-means showed a prediction accuracy of 79%, while the DBSCAN algorithm showed 100% accuracy of prediction accuracy.

3. SYSTEM DESIGN

The structure of the proposed laboratory risk classification system is shown in Figure 3. ① the current position information and the unique number of the researcher in the laboratory are transmitted to the safety management system. ②The safety management system stores the received data in the database. Refer to the in-memory cluster platform for the structure of the data stored in the database. ③ We implement and perform the clustering algorithm by combining Transformation and Action operations. Create risk and risk information based on the predictive model as output. ④ The researcher's real-time location information is compared with the prediction model, ⑤ and the notification and the danger information are transmitted when the risk is high. Finally, it is repeated ④ based on the new location record data.

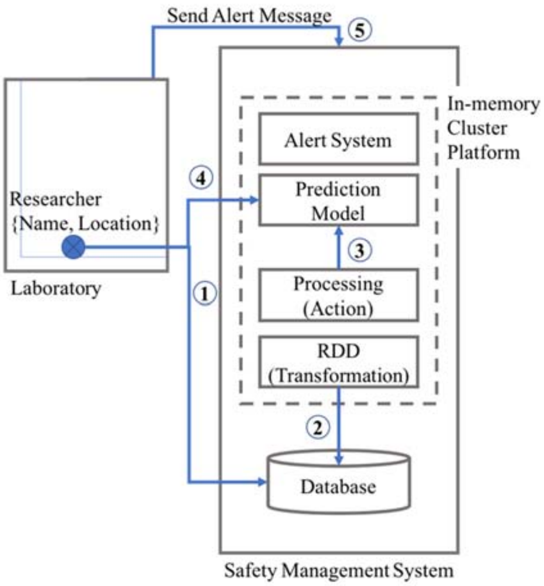


Figure 3: Laboratory Safety Management System

To quickly record and retrieve a large number of researchers' real-time location data, they use the NoSQL database system, which can store faster and more volume than traditional relational database systems. In addition, for the safety of the laboratory, the clustering algorithm operation for many location record data must be performed in a short time. Therefore, distributed processing and computation are performed using Apache Spark, an in-memory cluster platform that performs repetitive operations in memory fast.

3.1 Clustering Modeling System

Figure 4 shows the specific structure of the laboratory safety management behavior analysis and modeling system. The database system stores the location history data, the midpoint and risk of the prediction model, and key information. The clustering model has a central point for each group of clusters, and the risk for group risk factors at the nearest central point applies. The ideal center point location is located in the densest part of the location data contained in the group. Algorithm selection is important because the position of this central point changes according to the clustering algorithm.

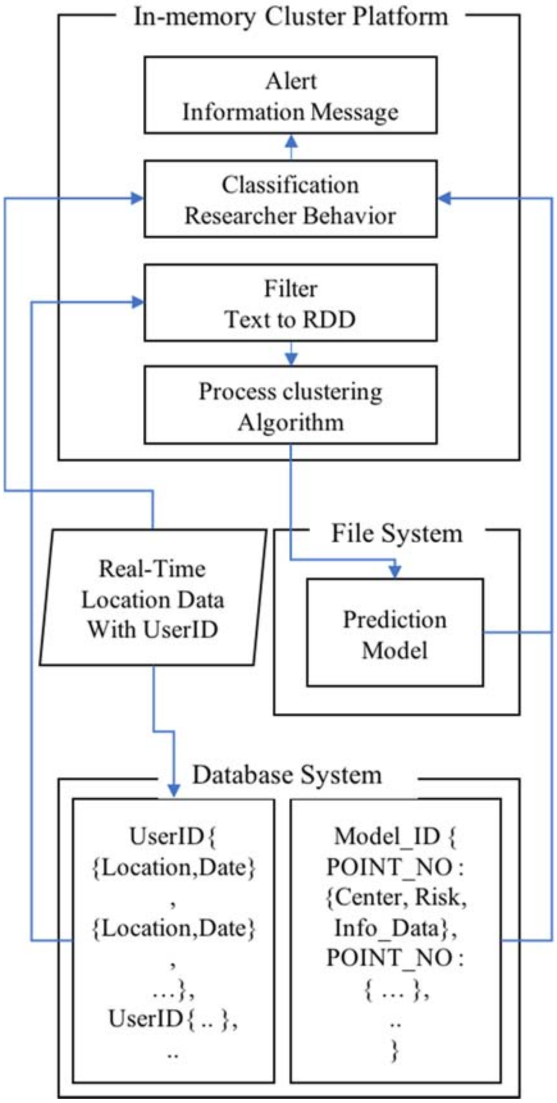


Figure 4: Clustering Modeling System

3.2 Location Classification Process

Figure 5 and Figure 6 is a flow chart for classifying the position data transmitted in real time in the proposed system. In order to quickly process the location information of the researcher, which is transmitted in real time, it is configured to open the socket and receive it by streaming. It converts the data type to Dense Matrix for computational efficiency. Spark supports the Dense Matrix and the Sparse Matrix in RDD to increase the efficiency of matrix operations. And assigns the converted position data to the prediction model to classify the groups. The risk and risk information for the group stored in the database is retrieved and retrieved. When the specific position data received continuously is compared with the learned model, a notification message is sent to the gateway to send a warning when the cluster is close to the center point and the risk of the group is high.

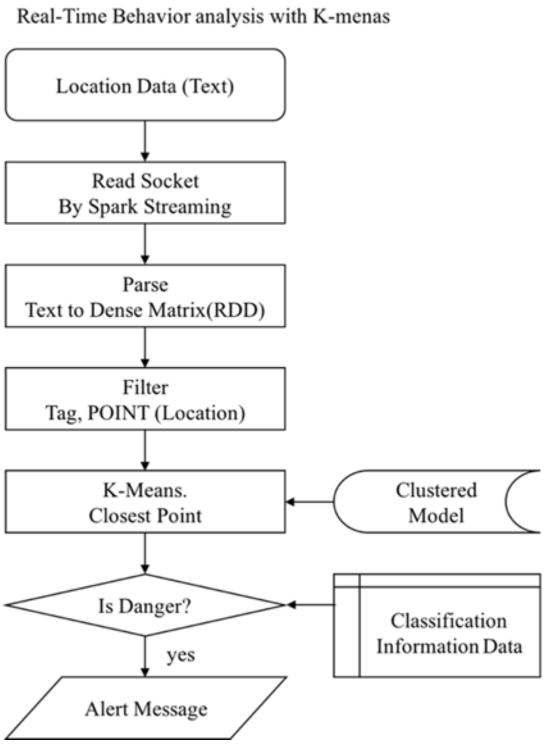


Figure 5: Real-Time Behavior K-Means Classification Analysis Flowchart

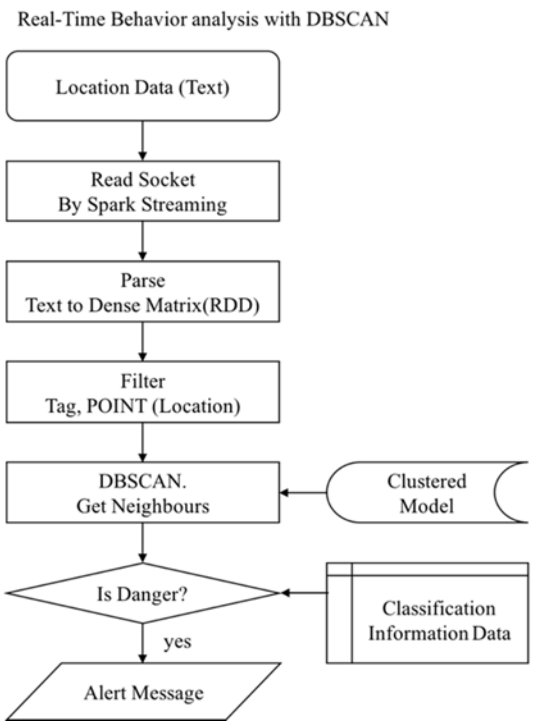


Figure 6: Real-Time Behavior DBSCAN Classification Analysis Flowchart

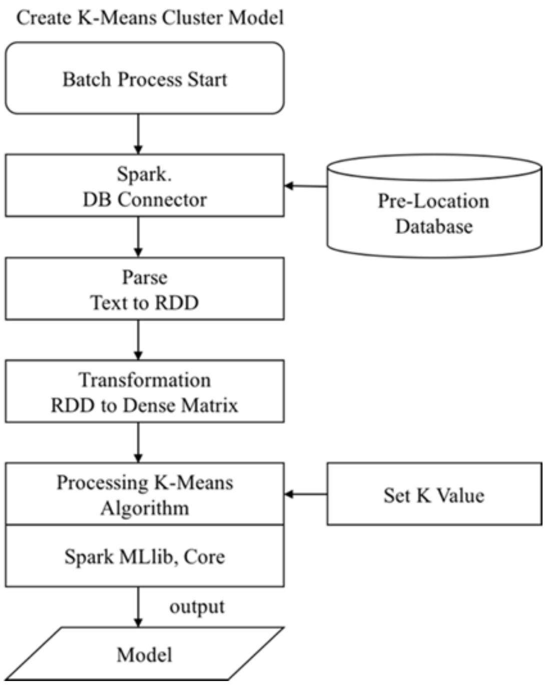


Figure 7: Create K-Means Clustering Model Flowchart

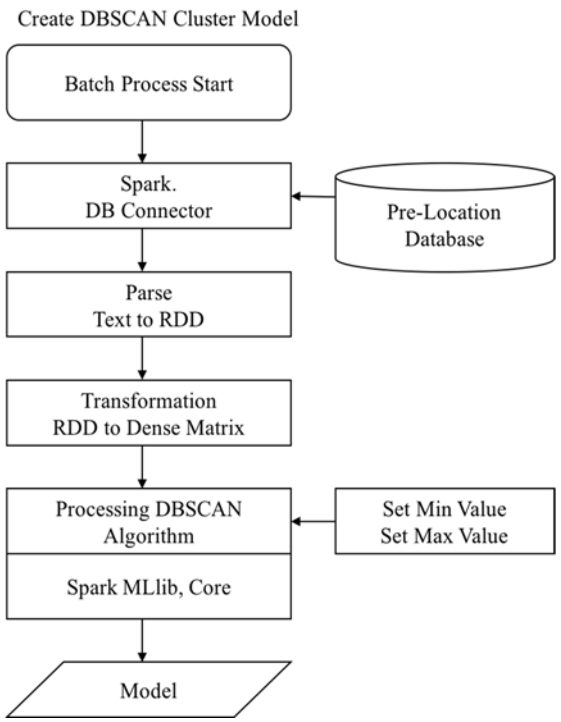


Figure 8: Create DBSCAN Clustering Model Flowchart

In order to analyze the behavior according to the user 's location, a pre-learned behavior prediction model is needed. To create a behavioral prediction model, the researcher's movement record should be able to classify specific locations and specify risk and risk information. Figure 7 and Figure 8 is a flow chart for creating a classification model in the proposed system. After importing the previous location record data stored in the database, convert it to the Dense Matrix form of RDD. It performs a K-Means and DBSCAN respectively converted Data Set. DBSCAN specifies the minimum number of groups and the maximum number of groups, and inserts the converted data set. As the classification algorithm is performed where there is data to configure different cluster and it is stored, which can be output to the final output model file. This model file can be used again as input and can also be used for real-time position data classification.

4. RESULT AND DISCUSSION

In this section, we investigate which classification algorithms are suitable for the proposed laboratory safety management system and improve of performance when using Apache Spark. In order to compare classification algorithms, we compared the programs implementing K-Means and DBSCAN clustering algorithms in Scala language in JVM environment. Second, when comparing Apache Spark with existing system, we implemented two algorithms in Apache Spark environment. In order to implement algorithms in Spark environment, you can use Scala or arithmetic libraries used in Java, but in order to use data converted to RDD type, it is necessary to use an appropriate arithmetic library. In this paper, an open source library is used to implement a clustering algorithm suitable for Spark environment.

The procedure of receiving the researcher 's position data in real time of the proposed laboratory safety management system in the paper can be implemented by spark streaming, but it is not implemented in the experiment. In addition, risk information and risk were not included in the calculation. It was not so unnecessary procedures There look out for better performance.

4.1 Location Data Set

The location record data set used in the experiment was modified from the GPS data set provided in [17] and the Gaussian data set in [18]. Are data having 6500, 13500, and 70,000 points (x, y), respectively. If you are importing data stored in MongoDB from Java or Scala, you should use the Connector library. This library is optimized for the Spark Connector engine, resulting in faster performance. In this paper, we use a method to retrieve the data stored in the text form in order to obtain only the computation time. Figure 10 is part of the file where the location data is stored. We removed unnecessary strings from the Date set to avoid being affected by the speed of character parsing. And the K-means result for position data is shown in figure 9.

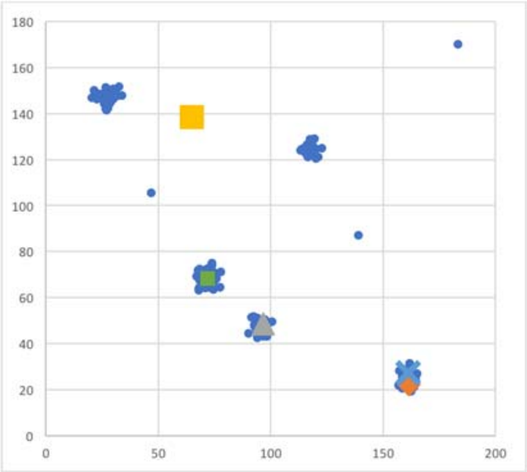
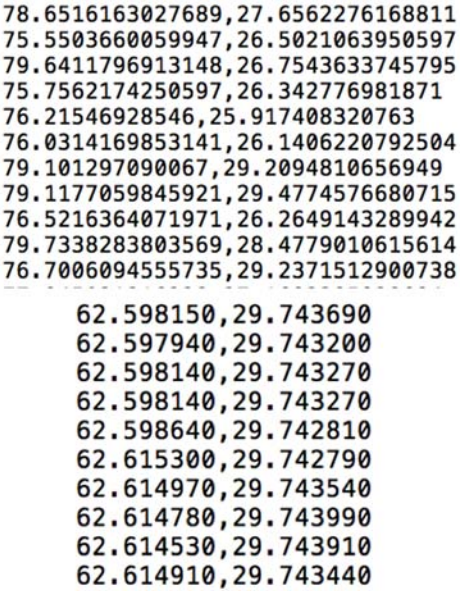


Figure 9: K-Means Clustering Result (k=5)



```
78.6516163027689,27.6562276168811
75.5503660059947,26.5021063950597
79.6411796913148,26.7543633745795
75.7562174250597,26.342776981871
76.21546928546,25.917408320763
76.0314169853141,26.1406220792504
79.101297090067,29.2094810656949
79.1177059845921,29.4774576680715
76.5216364071971,26.2649143289942
79.7338283803569,28.4779010615614
76.7006094555735,29.2371512900738
62.598150,29.743690
62.597940,29.743200
62.598140,29.743270
62.598140,29.743270
62.598640,29.742810
62.615300,29.742790
62.614970,29.743540
62.614780,29.743990
62.614530,29.743910
62.614910,29.743440
```

Figure 10: GPS Location Data Set

4.2 Experimental Environment

Experimental configurations are shown in Table 1 below. As a development tool, the IDE used IntelliJ IDEA 2017.1.3 64bit version. Table 2 shows the libraries used in the experiment.

Table 1: Experimental enviroments

Environment	Value
CPU	Intel Core i7 2.2GHz, 4 core
Memory	DDR3 1600MHz 16GB
Disk	SSD 256GB
OS	macOS Sierra

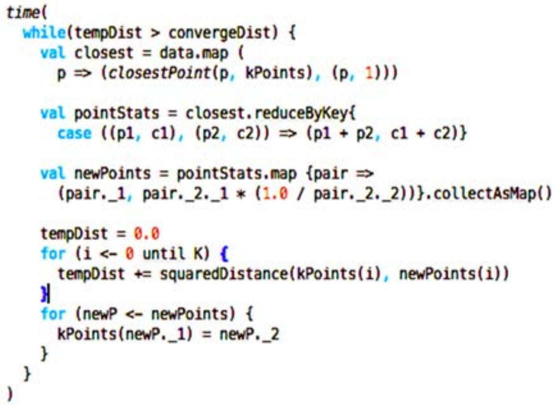
Table 2: Experimental Library

Platform / Library	Version
Apache Spark	2.1.1
Scala	2.11.8
JDK	1.8.0_131
MLlib	2.11
breeze	0.13.2
stark(dbis)	scala-2.11
NaK(scalanlp)	2.11:1.3

In a general Scala (JVM) environment, the K-Means algorithm uses the source code of the Apache Spark official Github page [14], and the DBSCAN algorithm uses the NaK library [15]. In the Apache Spark environment, the K-Means algorithm consists of MLlib and Spark Core, and the DBSCAN algorithm uses the open source library stark [16]. The library that loads the data in a normal Scala environment uses breeze. In order to load data in the general Scala environment, the file I / O function is used. In the Spark environment, the data stored in the text file is loaded into RDD through the breeze library used for numerical processing and converted into Dense Matrix.

4.3 Implementation

In this paper, we implemented the Scala language in the JVM to design the performance improvement of Apache Spark. We used System.nanoTime function to measure the exact time in our code. In Figure 11, the portion enclosed by the time function measures the execution time. The code was written according to the K-Means algorithm above. If you are using Spark, you can use the default classification algorithm of the MLlib library. Figure 12 shows the source code of a program that calls the K-Means function of MLlib in the Spark environment and outputs the result by inputting data and argument values converted to Dense Matrix. Because we need to use Spark's RDD, we use the ‘sc’ object, which is an abbreviation of Spark Context, to load the data. We used ‘time’ function in the line that commanded the operation to measure the operation time.



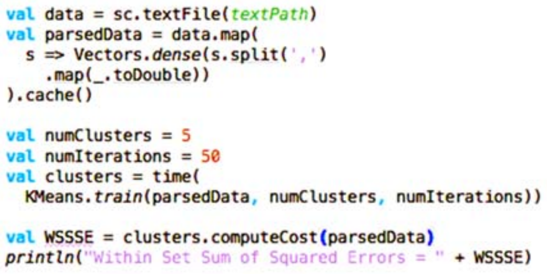
```
time(
  while(tempDist > convergeDist) {
    val closest = data.map (
      p => (closestPoint(p, kPoints), (p, 1)))

    val pointStats = closest.reduceByKey{
      case ((p1, c1), (p2, c2)) => (p1 + p2, c1 + c2)}

    val newPoints = pointStats.map {pair =>
      (pair._1, pair._2._1 * (1.0 / pair._2._2))}.collectAsMap()

    tempDist = 0.0
    for (i <- 0 until K) {
      tempDist += squaredDistance(kPoints(i), newPoints(i))
    }
    for (newP <- newPoints) {
      kPoints(newP._1) = newP._2
    }
  }
)
```

Figure 11: None-Spark K-Means Algorithm source code



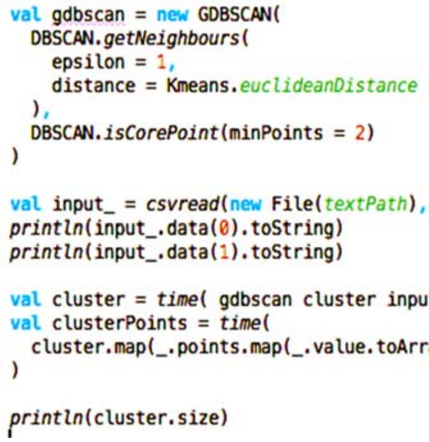
```
val data = sc.textFile(textPath)
val parsedData = data.map(
  s => Vectors.dense(s.split(',')
    .map(_.toDouble))
).cache()

val numClusters = 5
val numIterations = 50
val clusters = time(
  KMeans.train(parsedData, numClusters, numIterations))

val WSSSE = clusters.computeCost(parsedData)
println("Within Set Sum of Squared Errors = " + WSSSE)
```

Figure 12: Apache Spark K-Means Algorithm source code

In the case of DBSCAN algorithm, it is not supported in MLlib, so we implemented by referring to open source algorithm. Figure 13 is an integral part of the DBSCAN program source code without Spark. Apply the Euclidean distance to measure the similarity between data and specify the minimum number of sample data to avoid curse of dimensionality.



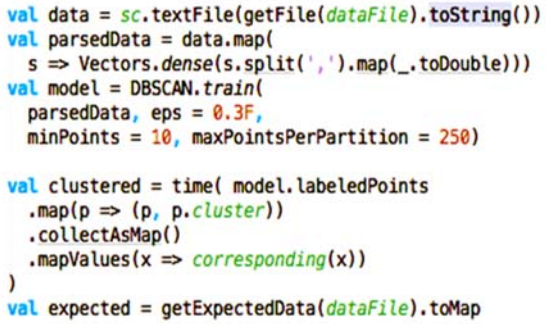
```
val gdbscan = new GDBSCAN(
  DBSCAN.getNeighbours(
    epsilon = 1,
    distance = Kmeans.euclideanDistance
  ),
  DBSCAN.isCorePoint(minPoints = 2)
)

val input_ = csvread(new File(textPath), ',')
println(input_.data(0).toString)
println(input_.data(1).toString)

val cluster = time( gdbscan cluster input_ )
val clusterPoints = time(
  cluster.map(_.points.map(_.value.toArray))
)

println(cluster.size)
```

Figure 13: None-Spark DBSCAN Algorithm source code



```
val data = sc.textFile(getFile(dataFile).toString())
val parsedData = data.map(
  s => Vectors.dense(s.split(',').map(_.toDouble)))
val model = DBSCAN.train(
  parsedData, eps = 0.3F,
  minPoints = 10, maxPointsPerPartition = 250)

val clustered = time( model.labeledPoints
  .map(p => (p, p.cluster))
  .collectAsMap()
  .mapValues(x => corresponding(x))
)
val expected = getExpectedData(dataFile).toMap
```

Figure 14: Apache Spark DBSCAN Algorithm source code

4.4 Comparison of Spark and None-Spark

Figure 7 shows the algorithm execution time when Apache Spark is not used and when using Apache Spark. The K-Means algorithm without Spark showed the lowest execution time of about 2000ms in 6500 and 13500 data sets. On the other hand, the execution time of the DBSCAN algorithm without Spark was the longest in about 8000ms. This is because the operation of DBSCAN is more complicated than that of K-Means, and more comparison operations are performed. However, when Spark was used, the execution time of all data sets from 6500 to 70,000 did not change significantly. Because demonstrate the high efficiency of a memory and the repetitive operation of more data in a distributed file system, if using a Spark.

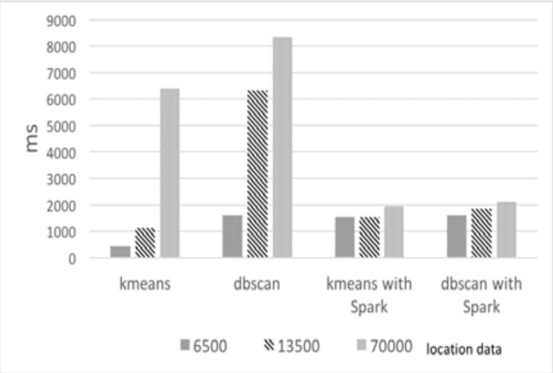


Figure 15: Spark and none-Spark Experimental Result Chart

Also, as shown in Figure 16, when DBSCAN algorithm is used, Spark shows 2 ~ 3 times faster performance. However, in the case of a small amount of calculations like K-Means, Spark seems to be slower. This is because the process of converting text data to RDD in Spark and the process of distributing and storing RDD are essential. Spark is inefficient for a small amount of simple processing.

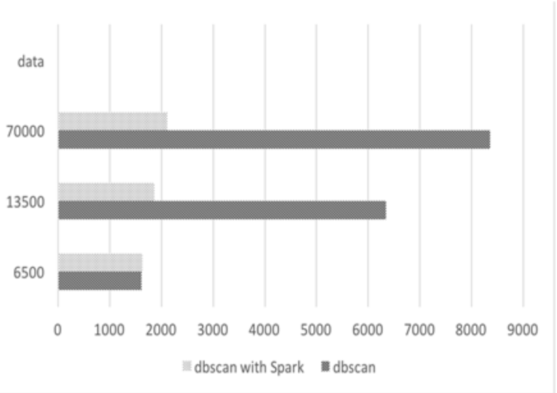


Figure 16: Spark and no-Spark Comparison chart

5. RELATED WORK

Since the existing laboratory safety management system research is based on static data analysis, it has a disadvantage that it cannot predict and prevent accidents that occur in real time. In addition, research on prediction of accident risk through real-time data classification algorithm has not been done yet.

In this paper, combining the clustering algorithm often used in traffic management for laboratory risk prediction model. The traffic management systems through classification algorithm finds an area where traffic accidents often happen to share the thinking type, or determine the risk in real-time. However, clustering algorithms are complicated and difficult to apply to safety management systems that predict and transmit dangerous situations in real time. Recently, an in-memory cluster platform suitable for such big data analysis is attracting attention.

The proposed system uses Apache Spark to analyze and manage the risk information and risk according to the researcher 's location. Dense Matrix type of RDD of Apache Spark core suitable for dense location data enables faster processing. We compared two more clustering algorithms to classify position data, and compared the results with those of existing systems.

As a result, Apache Spark showed faster processing speed and was not significantly affected by the amount of data. In the case of the existing system, the processing speed of the small amount of data was faster, but the processing speed was drastically decreased as the amount of the data increased. This is because of the characteristics of the classification algorithm, in which the same operations are repeated many times in proportion to the number of data. If you use a classification algorithm for a large amount of data processing such as a laboratory safety management system, you can get a speed improvement by applying Apache Spark.

6. CONCLUSION

In this paper, a situation classification and risk prediction system based on location data is designed and implemented for laboratory safety management. The proposed system can provide risk information and risk according to the location of the researcher using Apache Spark in real time. However, the scope of the system implemented in the paper is limited to the clustering and classification of location data. This is insufficient to deliver risk information and notifications to researchers in real time. In future research, we will design and construct more accurate prediction system by analyzing various kinds of data generated from various sensors and analyzing researcher's risk information and environmental information. We will also study IoT systems that can provide risk and risk information in real time.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2017R1D1A3B03036130).

REFRENCES:

[1] Author No.1, Author No 2 Onward, "Paper Title Here", *Proceedings of xxx Conference or Journal (ABCD)*, Institution name (Country), February 21-23, year, pp. 626-632.

[2] H.G. Lee, I.M. Lee, Y. T. Shin, J.Y. Moon, and I.B. Lee, "Development of Laboratory hazard discovery and management techniques",Journal of Korea Safety Management & Science,Korea Safety Management & Science, December, 2016

[3] Osang Jonathan Eyire, Obi, Emmanuel O, Ewona lgwe O, "Evaluations of the Effect of Workshop/Laboratory Accidents and Precautionary Steps towards Safety Practice", IOSR Journal of Electronics and Communication Engineering, May. - Jun. 2013, PP 16-22

[4] R. Li, H. Gao, D. Chu, K.Zhang, H.Xu, "Research on the Safety Accidents Prediction for Smart Laboratory Based on Statistical Analysis" ,IEEE ,May, 2017

[5] H. V. Jagadish, J. Gehrke, A. Labrinidis, et al., "Big Data and Its Technical Challenges", Communications of the ACM, July ,2014, pp,86-94

[6] http://hadoop.apache.org/, 7:48:14 GMT, 31/10/2017

[7] A. Svyatkovskiy, K. Imai, M. Kroeger, and Y. Shiraito, "Large-scale text processing pipeline with Apache Spark", Big Data (Big Data), IEEE International Conference on,December, 2016, pp. 3928-3935

[8] X. Meng, J. Bradley, B. Yuvaz, E. Sparks, S. Venkataramen, D. Liu, et al., "MLlib: Machine learning in apache spark", Journal of Machine Learning Research, 2016, pp. 1-7

[9] V.F. Agnes, H. Tamas, "Uniform data access platform for SQL and NoSQL database systems", Information Systems, September, 2017,pp. 93-105,

[10] Y. m. Sun, X.J. Liu, Q.Y. Sun, X.G. Chen, "Research and Application of Wireless Security Monitoring System in Chemistry and Chemical Engineering Laboratory",Wireless Personal Communications, August 2017,pp 2331–2344

[11] L. Gong, H. Sato, T.Yamamoto, T. Miwa, T. Morikawa, "Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines " Joural of Modern Transportation, September, 2015, pp 202-213

[12] R.Mauro, M.D. Luca and G.D.Acqua, "Using a K-Means Clustering Algorithm to Examine Patterns of Vehicle Crashes in Before-After Analysis" Modern Applied Science, October, 2013

[13]DOĞRU, NEJDET, and ABDÜLHAMİT SUBAŞI. "Comparison of clustering techniques for traffic accident detection." Turkish Journal of Electrical Engineering & Computer Sciences 23.Sup. 1 (2015): 2124-2137.

[14]https://github.com/apache/spark/blob/master/examples/src/main/scala/org/apache/spark/examples/LocalKMeans.scala, 7:48:14 GMT, 31/10/2017

[15] https://github.com/scalanlp/nak,

[16] https://github.com/dbis-ilm/stark, 7:48:14 GMT, 31/10/2017

[17] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems", Pattern Recognition, 39 (5), 761-765, May 2006.

[18] http://cs.uef.fi/mopsi/data/, 7:48:14 GMT, 31/10/2017