# Scalable Analysis of Social Media Logs Literature Review

Shang-Ping Dai
Computer Science Honours
University of Cape Town
Cape Town, South Africa
dxxsha001@myuct.ac.za

## ABSTRACT

Twitter generates large volumes of user generated data on a daily basis. An opportunity to extract valuable information arises to try and find patterns within this data and find interesting or unexpected related topics within communities or individuals in Twitter, but the problem of processing large volumes of data has been a problem that cannot be trivially done. This review will discuss techniques and methods that can help solve this scalability issue such as: retrieving a smaller subset of tweets to process with and compare efficient pattern recognition algorithms to process these small subsets of the data. These will be used to find frequencies and relations in the behaviours of communities or individuals on Twitter. The patterns found can be used in many ways such as finding related interests, to study behavioral changes or even used for targeted advertising to different communities. What is found in the discussed techniques is that efficiency can be achieved through many aspects when dealing with large scale data, such as the combining of different methods in the same field, as with the case of Information Retrieval, or by implementing standard algorithms different frameworks to reduce overall computation, as with the case of association rule mining techniques and implementations.

## KEYWORDS

Association Rule Mining; Information Retrieval; Mining; Parallelization; Cloud Computation

## 1 INTRODUCTION

Popular social media platforms have many active users that communicate with communities or with other individual users almost every day. Twitter is one such platform which contains massive amounts of user-generated content [24]. It is estimated that on average, around 6000 tweets are tweeted every second [24]. This makes it extremely difficult for researchers or scientists to process such vast amounts of data to obtain useful information. This struggle has been expressed by research [22] that deals with large data.

In this project, we wish to explore the implementation of efficient methods to find patterns that relate to a user's interests in Twitter posts to understand user behaviour on a micro and macro level. These patterns of user interest can have a variety of uses that range from targeted advertising to more relevant search results and even tweet or user recommendations to keep a user interested and invested in the platform.

This problem can be roughly divided into two parts, namely, information retrieval (IR) of useful twitter data on specific targeted areas and the recognition of patterns in the retrieved data to find frequent or interesting patterns in the data.

These two steps were designed to solve the two high level categories of challenges faced when faced with processing large amounts of social media data, identified and discussed briefly by Muhammed et al [1]. They say that these challenges can be identified as scalability issues, and content issues. Scalability issues refers to the amount of data that would need to be processed to obtain relevant information and content issues refer to the quality of user content of the social media platform. Tweets on Twitter are known to be more conversational and closer to speech than actual writing. This may be problematic when attempting to process tweets to find patterns as implications in tweets might infer different meanings that what is literally presented [1].

Scalability issues have been a long-time problem for researchers and scientists when wanting to process data. Although many methods exist to deal with this issue, Armbrust et al [25] discuss how Cloud Computing addresses scalability by splitting computation on multiple machines. They argue that "batch-oriented tasks can get results as quickly as their programs can scale, since using 1000 servers for one-hour costs no more than using one server for 1000 hours". Cloud Computing is also seen to be used as a viable solution by much research in finding patterns in large datasets [18, 21].

This literature review will identify and provide a critical overview of current research done in areas that are applicable in the context of the project. The literature review will also touch on the current standards of performance in each field and challenges faced. This will lead to a discussion of which methods could potentially be applied to the context of the project and an analysis of the current standard in related fields.

Finally, the conclusions section will summarize reviewed papers and work done in related fields.

## 2 INFORMATION RETRIEVAL

Heimstra [3] states that a basic Information Retrieval system should support three basic processes: the representation of the content of documents, the representation of the user's information

need and the comparison of the two representations. The representation of documents is also referred to as the *indexing* process. This is often where the documents are changed to an index representation, such as converting all text to lower case. The indexing process may also include the partial storage of the document. The representation of user needs refers to the *query formulation* process, in which the user's information need is given to the system as a suitable query to use to fetch documents. Finally, the *matching* process compares the query against document representations to present a ranked list of documents to the user. It is expected of an IR system to do ranked retrieval in the matching process [2]. The notion of ranked retrieval refers to ordering the list such that the documents that were the most relevant to the user's query would be at the top and less relevant ones would follow. Various models of IR were developed to quantify this relevancy to the user's query.

## 2.1 IR Models

In the field of IR, there exists many models [3] to aide as a blueprint to guide the implementation of an IR system. Early IR systems started with Boolean models in which users specified their information needs using a complex combination of boolean ANDs, ORs and NOTs; such entries that simply adhered to the query would be returned. This is disadvantageous as no form of ranking could help the user determine the most relevant documents retrieved.

Singhal et al [2] mentions that the three most used models in IR research are the vector space model, the probabilistic model, and the inference network model.

The vector space model [2, 3] is used for ranked IR by representing text as a vector of terms (which are typically words or phrases). A new dimension is then made for every word in the vocabulary (which may consist of millions of terms), which results in a high-dimensional vector space. Documents are then made into text-vectors in this space and assigned a non-zero value if it corresponds to terms in the vocabulary. A query vector would then be made as a text-vector of the query in this space and similarity could be scored with the document vectors in the space with the query vector. This is done by using the cosine of the angle between the two vectors, which is the dot product of the two vectors.

The probabilistic model [2, 3] retrieves and presents documents to the user in decreasing probability of whether the user will find them useful or not. This is called the probabilistic ranking principle (PRP) by Robertson [4]. The PRP was found off the basis that the IR system is never *certain* about what the user would find useful and thus a system should be made to determine a way to quantify how *probable* a user would find the document useful. A probabilistic model ranks documents by estimating the probability of relevance of documents; probabilistic IR models differ with this key estimation.

The inference model [2, 3] uses an inference network to make decisions on which documents to retrieve based on a network of knowledge. Each document in the network will instantiate a term with a certain strength. A score (of relevance) is accumulated when a query is given, and the highest scoring documents will be retrieved for the user.

These models, and more have all been developed to find new and more efficient ways to retrieve archived data.

## 2.2 Effectiveness of IR Implementations

Objective evaluation of IR systems is a challenging part of the field of IR. The research community have agreed that a good IR system should be able to retrieve as many relevant documents as possible and will retrieve as few non-relevant documents as possible [2]. This is classified as: recall - the proportion of relevant documents that are retrieved; and precision - the proportion of documents retrieved that are relevant [6]. Many in the field use these quantifiers as metrics in evaluation studies for IR systems, such as research done to evaluate search engines to retrieve scholarly information done by Shafi et al [7]. These two metrics have been weighted against one another over the years as techniques that improve recall have tend to hurt precision and vice-versa.

Using IR systems, a smaller subset of data can be retrieved to be processed to find patterns within the data. This can be done through various pattern recognition techniques.

# 3 PATTERN DETECTION

## 3.1 Association Rule Mining

Data mining is a well-researched topic in Computer Science. Its aims are to find interesting correlations and frequent patterns [10] (usually) in transactional databases or other data repositories. Association rule mining is a type of data mining technique that makes use of building a set of rules (usually very large) from a dataset. These rules are constructed using a minimum *support* and *confidence* level. The support level is used to determine itemsets that are large or frequent to make rules from. The confidence level is a threshold entailing that we must only accept a rule as valid if the rule applies for at least a certain percentage of the dataset. The rules that are found describe a relation between two sets of items. This indicates that for some transaction in the dataset, if the transaction contains an item from the first set, it is likely to contain an item from the second set.

Association rule mining has had deep applications in transactional databases, where patterns can be found in customer transactions that could benefit a business by identifying related items that are always purchased together [13].

## 3.2 Algorithm Structure

Kotsiantis et al [10] describes in their overview of association rule mining that it can be split into two subproblems. The first is to find itemsets whose occurrences exceed a predefined threshold (the support level) in the dataset (these are called *frequent or large itemsets*), and the second subproblem is to generate association rules from those large itemsets with the constraints of minimal confidence.

They mention that the generation of association rules is straightforward, so researchers rather focus on the generation of

frequent itemsets which is done by splitting it into two further subproblems. These are the generation process of the candidate large itemsets, and the generation process of the frequent itemsets. The candidate itemsets are those itemsets in which there is hope that it would become frequent itemsets, and the frequent itemsets are itemsets whose support exceed the support threshold (i.e they appear frequently enough in the dataset).

They also mention in their overview that the rules generated are often on the order of thousands to millions. Research in the field has often been to propose several strategies to reduce this number, such as only generating "interesting" rules, only generating non-redundant rules or only generating rules that adhere to certain criteria such as lift, strength, coverage or leverage. It was also described by Hipp et al [8] that the problems they identified were attempting to prune the set of rules that the system would generate (as often, many of the rules would be redundant) and that only "interesting" rules should be further selected from the set of generated rules. These interesting rules could be found by iteratively checking the rules with updated confidence levels in each iteration.

Kotsiantis et al also discusses that progress to increase the efficiency of the algorithm has been made in the following four directions: sampling the database [13], adding extra constraints to the structure of patterns [14, 9], parallelization of different aspects of the mining algorithm [13] and reducing the number of passes over the database [15].

## 3.3 Efficiency

Many researchers advance the field of data mining by finding new methods of reducing computation or to add extra constraints in which to prune rules mined from the data.

One method of reducing computation time is to sample the database into partitions of non-overlapping sets which could be worked on separately [13]. This is used as an approach to solve the first subproblem introduced previously (identifying the large itemset). The partitioned sets would each have their own *local* large and candidate itemsets, each partition would also need its own *local* support threshold since the partition would include only a small subset of the database. This notion was also discussed in Kotsiantis et al's [10] overview, where they mentioned that not all rules could be found on the first pass of these partitions/samples and some associations would be deemed infrequent in the partition but could actually be frequent in the full database. The union of all local large itemsets would then be the global large itemset of the database. This approach was taken in the early days of the field by Savasere et al [13] and provides many advantages, such as the parallelization of the step, which they had covered as future work that could be done. Their partition algorithm approach was tested with the Apriori algorithm, which was the standard for association rule mining and concluded that it had CPU and I/O improvements over Apriori. It was also found that this partition algorithm scaled linearly with the problem size.

Other work done by researchers [11, 26] focused on areas in which using convenient data structures on association rule algorithms could lead to visible improvements in computation time. This was often due to identifying parts of the standard Apriori algorithm that were performing redundant processing and using a data structure to improve on those specific areas.

Research was also done to improve association rule mining by introducing pruning techniques to avoid measuring certain itemsets while guaranteeing completeness of the rules generated [9]. This was termed remaining tuple optimization and could be used to prune off "new" candidate itemsets as early as it is generated. Their results showed that while their optimization technique could improve estimation accuracy, the algorithm would have to make more passes over the database, which would result in longer computation times.

The methods and efficiencies discussed help the standard Apriori in specific areas decreasing computation by visible amounts. However, these algorithms would still struggle in processing large volumes of data present today.

## 3.4 MapReduce Model

Recent interest in the data mining field has shifted to using the MapReduce framework [23] to implement association rule algorithms. Widely used in Google, Dean et al [23] states that the model has many advantages. Firstly, it is an easy to use model for programmers without experience in distributed and parallel systems by hiding the details of its parallelization. Secondly it has many applications to various areas, including data mining. Lastly, the implementation of MapReduce can scale to large clusters of machines, which can make efficient use of machine resources if available. These advantages have been used by many researchers to implement this model in the data mining field [22, 20, 19].

In brief, the entire computation takes a set of input key/value pairs and produces a set of output key/value pairs. Well-defined intermediate results are passed so that implementation of the framework is split into two functions: map and reduce.

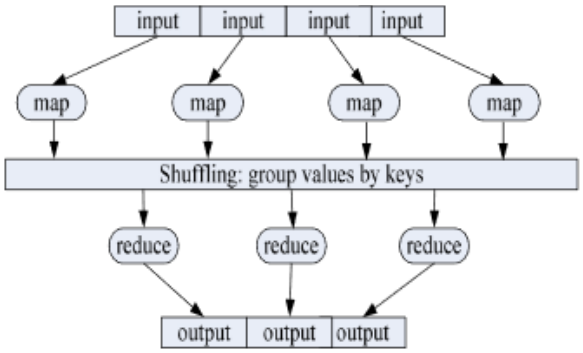

Figure 1: MapReduce framework

In figure 1, the illustration shows the MapReduce framework. The map function is invoked on all inputs (partitions) which generates intermediate results for the reduce function to process to provide the output. These map and reduce invocations can be easily parallelized to be called on multiple machines. The application of this model is where much research on increasing the efficiency of association rule mining is done [22, 20, 19].

As discussed previously, the standard Apriori algorithm has two steps: the generation of frequent itemsets and the generation of association rules. Implementations of Apriori onto the MapReduce framework have been approached by implementing one of these two steps of Apriori onto the map and reduce functions respectively. Usually the generation of frequent itemsets would be implemented as the generation of association rules is straightforward given the frequent itemsets.

The MapReduce framework has provided visible improvements to computation time to the standard association rule algorithm [21, 27, 18] especially since the model provides a good structure to implement and use in a cloud computing environment.

### 3.5 Cloud Computing

Li and Zhang [18] describe cloud computing to be a form of computation that distributes a number of tasks to a resource pool consisting of a large number of computers. Cloud computing provides cheap, aggregated storage space and computing power over multiple systems that help create a desirable platform for the storage and mining of data [21].

The structure of association rule cloud computing algorithms is similar to parallelized MapReduce or Apriori algorithms. The key difference to the two is that in the cloud computing framework there exists a notion of a *master node* [18, 21]. This node acts as a job-tracker and is responsible for: scheduling jobs that the other nodes are expected to process, monitor all nodes and re-execute or invoke tasks in the case of failed tasks. It was also highlighted that the key to take advantage of a cloud computing environment adequately was the partitioning and distribution of the dataset to different processing nodes. This is to relieve the impact of load balancing and communication granularity.

In many papers, researchers [21, 27] use the Hadoop software library framework to implement distributed association rule algorithms using a MapReduce model. Hadoop is a free, Java-based programming framework that supports distributed processing on large numbers of machines that do not share any memory or disks. It creates clusters of machines and coordinates tasks each node does in the clusters. Hadoop is said to consist of two key services: reliable data storage using their own Hadoop Distributed File System (HDFS) and high-performance parallel data processing using MapReduce. Hadoop also uses replication to avoid data loss if multiple nodes fail, Hadoop also keeps tracks of the locations of replicated files.

## 4 DISCUSSION

Systems that are expected to process large volumes of data are not trivial to implement, however many advances in the field prove that it is not impossible. As seen in the research discussed, many areas may overlap to form an efficient solution to a complex problem.

In the field of Information Retrieval, IR systems use different models to structure algorithms to fetch relevant documents based on a specific query. The effectiveness of these results is also measured by metrics called recall and precision. Work had been done by Becker et al [17] to retrieve social media information about upcoming events that would be happening. They used an algorithm that included both recall-orientated and precision-oriented query generation techniques which the paper concluded to provide improved results to the documents retrieved by their system.

It is mentioned by Heimstra et al [3] that there is no such thing as a dominating model in IR, each model has it's own applications in different circumstances. Vector space models have been found to be well suited for similarity search and relevance feedback in many (non-textual) situations if a good weighting function is available and probabilistic models are a good choice if examples of relevant and non-relevant documents are available.

As discussed in section 1, systems that process data to obtain information about the data will often struggle with content issues when dealing with social media data. User-generated content on such social media platforms are written in a natural language context. A post that comments on a certain topic may infer other meanings than just its literal meaning, but more fundamental than that, it is difficult to truly know whether a machine understands human meanings. These problems are touched on by Allen et al [16] on a higher level. They discuss the topics in the field of natural language processing, listing ambiguity principles that machines or systems cannot easily assign meaning to. Natural language processors that derive meaning from ambiguous test are called semantic interpreters. These semantic interpreters are on the edge of the scope of the project, but a simple implementation might be considered to increase the quality of the rules that are found. The results of the research discussed showed that inference network models perform better than conventional probabilistic models and the use of multiple query representation can be used to further enhance performance.

Following the retrieval of items, we are then presented with a subset to process further. We wish to find patterns or rules within this set that describe patterns in a user or community's behavior. The various approaches done by researchers to implement association rule mining algorithms have made progress in increasing the efficiency of the standard Apriori algorithm using various techniques.

Early association rule mining algorithms made advances by partitioning the dataset [13]. This Partition algorithm brought great improvements to the standard Apriori algorithm getting faster computation times and less comparisons that needed to be made. Other work around the same time made use of complex data structures (use of clustering schemes based on equivalence classes and maximal hypergraph cliques) [15]. The use of complex data structures gave the need to only make one pass through the database which is where they obtained efficiency increases. These algorithms have also shown visible improvements to the Partition algorithm by having an order of magnitude less computation time, as results from research have shown [15]. Even with these computation improvements, these algorithms had been evaluated with fairly small amounts of data and have not been implemented on the large volumes of data present in many of today's systems.

Interest in the field has thus shifted to using cloud computing to make use of large clusters of commodity computers connected

together. This could be done with the use of the MapReduce framework discussed in section 3. As discussed in research that have implemented Apriori over the MapReduce framework on cloud computing platforms [18, 27, 21, 19], computation times have increased drastically compared to older solutions when using data many magnitudes larger. The addition of complexity to the standard implementation of Apriori over MapReduce can also see visible improvements. This was done by Farzanyar et al [22] by adding pruning techniques to reduce computation by pruning unneeded partial frequent itemsets in the generation of frequent itemsets process.

Cloud computing also brings the added benefits that cloud computing brings such as: cost-efficient solutions to storing and analyzing mass data and simple scalability of large problems.

**Table 1. Summarised effective data mining techniques and their results**

| Description | Platform | Results |
|---|---|---|
| Partition algorithm that splits the database into partitions for the generation of the frequent itemset (sequential) [13] | Single core machine | Linear scaling with the problem set, performed better than standard Apriori, reduced CPU and I/O overhead, good framework for parallelization |
| Use of complex data structures to make use of lattice traversal techniques [15] | Single core machine | Only one pass is needed through the database, significant improvements in computation time |
| Apriori's generation of frequent itemsets over MapReduce [18, 21, 27] | Apache's cloud computing framework, Hadoop | Dramatically improved computation time for datasets two orders of magnitude larger than the ones used in the first two techniques |
| Apriori's finding of association rules over MapReduce [19] | Apache's cloud computing framework, Hadoop | CPU and I/O scale up well with the dataset, but computation time has sublinear performance, speed up not good for small datasets (1GB), does better for large datasets (8GB) |

## 5 SUMMARY

This review has explored and elaborated on potential solutions to the problem of finding frequent patterns of Twitter users on a macro and micro level with the usage of Information Retrieval techniques to obtain a smaller subset of a large volume of data and then further generating rules that relate phrases or words together using efficient association rule mining algorithms to infer behavior or related topics discussed on Twitter.

The research considered in this review expressed brief overviews of challenges faced in each approach and which specific areas could be more efficiently done. This was done by using similar techniques or by adding extra complexity to a subproblem of the goal, which was the case of association rule mining algorithms, or by combining techniques used in the same field to yield better results, as in the case of combining different models in IR to retrieve documents.

It is also seen by prior work that implementing the standard Apriori algorithm on the MapReduce framework to use in a cloud computing environment can solve the scalability issue of finding association rules on large volumes of data. In particular, the parallelization of the frequent itemset generation process over the MapReduce framework can help with the scalability of the problem set. Although more visible improvements to computation time can be achieved through the use of complex data structures or added complexity to the standard algorithm (such as pruning techniques), the ease of implementation should also be considered and balanced accordingly.

## REFERENCES


[1] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing social media messages in mass emergency: A survey. ACM Comput. Surv. 47, 4, Article 67 (June 2015), 38 pages. DOI: http://dx.doi.org/10.1145/2771588

[2] Singhal, A., 2001. Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4), pp.35-43.

[3] Hiemstra, D., 2009. Information retrieval models. Information Retrieval: searching in the 21st Century, pp.1-17.

[4] Robertson, S.E., 1997. The probability ranking principle in IR. *Readings in information retrieval*, pp.281-286.

[5] Howard Turtle, W. Bruce Croft, 1991. Evaluation of an inference network-based retrieval model,ACM Transactions on Information Systems (TOIS), v.9 n.3, p.187-222. DOI: http://dx.doi.org/10.1145/125187.125188

[6] Dumais, S.T. 1991, June. Improving the retrieval of information from external sources. *In Behavior Research Methods, Instruments, & Computers (1991)* 23 (2), (pp. 229-236). DOI: http://doi.org/10.3758/BF03203370

[7] Shafi, S.M. and Rather, R., 2005. Precision and recall of five search engines for retrieval of scholarly information in the field of biotechnology. http://dspaces.uok.edu.in/jspui//handle/1/198

[8] Hipp, J., Güntzer, U. and Nakhaeizadeh, G., 2000. Algorithms for association rule mining—a general survey and comparison. ACM sigkdd explorations newsletter, 2(1), pp.58-64. DOI: http://doi.org/10.1145/360402.360421

[9] Agrawal, R., Imieliński, T. and Swami, A., 1993, June. Mining association rules between sets of items in large databases. In Acm sigmod record (Vol. 22, No. 2, pp. 207-216). ACM. DOI: http://doi.org /10.1145/170035.170072

[10] Kotsiantis, S. and Kanellopoulos, D., 2006. Association rules mining: A recent overview. GESTS International Transactions on Computer Science and Engineering, 32(1), pp.71-82.

[11] Margahny, M.H. and Mitwaly, A.A., 2005, December. Fast algorithm for mining association rules. *In the conference proceedings of AIML, CICC*, pp (36-40) Cairo, Egypt (pp. 19-21).

[12] Parthasarathy, S., Zaki, M.J., Ogihara, M. and Li, W., 2001. Parallel data mining for association rules on shared-memory systems. Knowledge and Information Systems, 3(1), pp.1-29.

[13] Savasere, A., Omiecinski, E.R. and Navathe, S.B., 1995. An efficient algorithm for mining association rules in large databases. Georgia Institute of Technology.

[14] Toivonen, H., Klemettinen, M., Ronkainen, P., Hätönen, K. and Mannila, H., 1995. Pruning and grouping discovered association rules.

[15] Zaki, M.J., Parthasarathy, S., Ogihara, M. and Li, W., 1997, August. New Algorithms for Fast Discovery of Association Rules. In KDD (Vol. 97, pp. 283-286).

[16] Allen, J.F., 2003. Natural language processing. *Encyclopedia of Computer Science, 4th Edition*, pp.1218-1223.

[17] Becker, H., Iter, D., Naaman, M. and Gravano, L., 2012, February. Identifying

content for planned events across social media sites. *In Proceedings of the fifth ACM international conference on Web search and data mining* (pp. 533-542). ACM.
DOI: http://doi.org/10.1145/2124295.2124360

[18] Li, L. and Zhang, M., 2011, July. The strategy of mining association rule based on cloud computing. *In Business Computing and Global Informatization (BCGIN), 2011 International Conference on* (pp. 475-478). IEEE.
DOI: http://doi.org/10.1109/BCGIn.2011.125

[19] Li, N., Zeng, L., He, Q. and Shi, Z., 2012, August. Parallel implementation of apriori algorithm based on mapreduce. In *Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on* (pp. 236-241). IEEE.
DOI: http://doi.org/10.1109/SNPD.2012.31

[20] Lin, M.Y., Lee, P.Y. and Hsueh, S.C., 2012, February. Apriori-based frequent itemset mining algorithms on MapReduce. In *Proceedings of the 6th international conference on ubiquitous information management and communication* (p. 76). ACM.
DOI: http://doi.org /10.1145/2184751.2184842

[21] Li, J., Roy, P., Khan, S.U., Wang, L. and Bai, Y., 2012, December. Data mining using clouds: An experimental implementation of apriori over mapreduce. In *Proc. 12th International Conference on Scalable Computing and Communications (ScalCom'13)* (pp. 1-8).

[22] Farzanyar, Z. and Cercone, N., 2013, August. Efficient mining of frequent itemsets in social network data based on MapReduce framework. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 1183-1188). ACM.
DOI: http://doi.org/10.1145/2492517.2500301

[23] Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), pp.107-113.
DOI: http://doi.org/10.1145/1327452.1327492

[24] Internetlivestats.com. (2018). *Twitter Usage Statistics - Internet Live Stats*. [online] Available at: http://www.internetlivestats.com/twitter-statistics/ [Accessed 28 Apr. 2018].

[25] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., 2010. A view of cloud computing. *Communications of the ACM*, *53*(4), pp.50-58.
DOI: http://doi.org /10.1145/1721654.1721672

[26] Park, J.S., Chen, M.S. and Yu, P.S., 1995, December. Efficient parallel data mining for association rules. In *Proceedings of the fourth international conference on Information and knowledge management* (pp. 31-36). ACM.
DOI: http://doi.org /10.1145/221270.221320

[27] Yang, X.Y., Liu, Z. and Fu, Y., 2010, June. MapReduce as a programming model for association rules algorithm on Hadoop. In *Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on* (pp. 99-102). IEEE.