IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingenierie des Systemes d'Information
Vol. 24, No. 1, February, 2019, pp. 51-56
Journal homepage: http://iieta.org/Journals/isi

A Novel Frequent Pattern Mining Technique for Prediction of User Behavior on Web Stream Data

Pandluri Dhanalakshmi

Department of CSSE, Sree Vidyanikethan Engineering College, A. Rangampet 517102, Tirupathi, India

Corresponding Author Email: mallidhana5@gmail.com

https://doi.org/10.18280/isi.240107

Received: 3 October 2018
Accepted: 11 January 2019

Keywords:
frequent pattern mining, classification, user behavior, web data, data extraction

ABSTRACT

In recent years, as the size of the online databases increases, content in the web pages also increases, then the human behavior towards the online content has become a major issue for decision making. Hence, the need for extracting knowledge from high dimensional online content using automated techniques also increased. To address the decision making issues, a novel dynamic model is required to discover the required knowledge from these databases. In this paper, a novel filter based user navigation pattern mining model is designed and implemented on the large online steaming databases. Experimental results proved that the present filtered based frequent pattern mining model efficiently predicts the user navigation patterns with high accuracy and less runtime.

1. INTRODUCTION

Over the last two decades in India due to rapid growth of E-Commerce, penetration of mobile phones has changed the way to communicate and do business through an Internet [1]. The main reasons for accessing the internet are entertainment, online communication and Social Networking. As per the information revealed by Google India in 2016 hundred million people are going online to purchase because of increasing of internet penetration.

One of the example for E-Commerce is Online shopping. Now a days over the internet most of the customers are directly buying the products.

Popular E-Commerce sites in India are:
Flipkart
Amazon
Snapdeal

To tack any communication and to record each transaction of the HTTP Protocol we need one file which is called as Web log file [2]. A weblog file is a file which records all user activities occurred over a period of time [3].

It also contains effective usage of web data. There are 4 types of Server logs: Transfer, Agent, Error and referrer log. If the number of customers increased from million to billion the size of log data also increases due to more transactions on online data [4]. Generally web usage data contains the information about IP Address of the user, the

type of file he accessed, which browser he is accessing, how many bytes he accessed [5].

This log file is also used to analyze the customer behavior like number of people who have visited the web site in how many sessions, Total time spent on a web site, how many times he accessed the same webpage [6].

Analyzing the customer behavior is the key factor to optimize the any E-commerce sites, to know the kinds of goods and problems they are facing during online shopping. Usage of internet based applications increases due to increased number of customers which will lead to increasing of event access log size produced on web servers [7]. The size of the access logs varies from Tera bytes to Peta Bytes.

For Periodic analysis of access logs software companies spent a large budget on development of these applications [8]. To handle event access logs efficiently in offline mode a traditional data analytic method called Hadoop provide the capabilities using Map Reduce Algorithm.

Hadoop is not an efficient Technique to handle the Unbounded and Streaming of log data as they produced in real world applications. The main drawback of offline processing is that some context and information collected from event log becomes useless and irrelevant [9].

A modified framework is designed to perform Parallel Processing of Big data than the Hadoop is called Spark. It permits both batch and stream Processing while Hadoop mostly for batch processing [10]. Even though Spark uses Hadoop Distributed File System (HDFS) in some circumstances Spark runs hundred times faster than Hadoop.

In addition to handling the static data when it is available Spark also process Real time data Applications within a short time [11]. Spark support various applications like Graph Computation, Processing of streaming data and in Machine Learning and it Perform analytics on these applications is very simple way and fast [12].

2. RELATED WORK

Over the last two decades, to process the large volumes of data several frameworks and models have been developed among which one of the most widely used technique is Hadoop Map Reduce. To handle interactive and dynamic data some of the framework model process the data in the form of batch processing [13]. Even though these framework handle today's use cases there is a need for processing large volumes of data with in a short time such techniques are called as Data streaming techniques [14].

To perform or to handle unbounded and continuous stream of data as they arrived in online a commonly used technique is called as stream processing [15]. Data stream processing are

also used to storage of data, for visualizing the results to analysts. Data streams also supports several real time database storage architectures such as NO SQL databases, relational databases, in-memory databases [16]. Some of the cloud providers provide data storage solutions are Google, Azure, Amazon.

Analyst use the result of data processing solution for interfacing web based API and visualization of presenting results [17]. Depending on the type of application, the data in the data stream includes video, time series data and event logs etc. To process dynamic or online or streaming data several systems have been developed, this type of systems are also called as Data Stream Management systems. To perform relational operations like join, aggregation and filtering operations in the table Data Stream Management system (DSM) is the most suitable technique which is also responsible for disk resident data [18]. To perform large queries, DSM also provide some declarative languages and CEP Systems on unbounded streams of data [19].

To support an efficient computations like interactive queries and stream processing, Apache spark extends event map reduce model as clusters computation [20]. Execution of these computations in memory in fault tolerant manner spark introduces a resilent distributed datasets for different workloads [21].

RDD is a programming interface for performing filter, map and join operations which are immutable. proposed a new method called as Data stream which handles the faults more efficiently and it is based on spark streaming [22]. This method periodically performs the stream processing in batch computations over the internet edges and transferring the events to the cloud in batches.

To perform operations and design online services, analyzing the user behavior is an important measure. To characterize the user activities in online databases and editing patterns in Wikipedia and to study the user's search intent recent works analyze the server logs and the network traffic in social networks.

Earlier research work was carried out to analyze the user behavior in online databases is called click stream analysis [23]. To find this behavior and user navigation paths, most of the researchers applied statistical methods like Markov chains and regression techniques chi-square tests.

These models are mainly useful for calculating simple aspects of user behavior means finding influence of any webpage for all users. After application of any one of clustering techniques to find the similar click stream activities for all users, the resulting clustering objects or clusters are used to predict the future user behavior or user interests [19].

In order to improve the amount of data transferred from sources to the cloud and to analyze the behaviour of users an efficient novel technique is developed in this paper.

3. PROPOSED MODEL

Even though Spark uses Hadoop Distributed File System (HDFS) in some circumstances Spark runs hundred times faster than Hadoop [24]. In addition to handling the static data when it is available Spark also process Real time data Applications within a short time.

Spark support various applications like Graph Computation, Processing of streaming data and in Machine Learning and it Perform analytics on these applications is very simple way and fast.

Spark is well suited one which is shown in Figure 1. The two important sub-components of the spark framework are:

Spark Streaming Core: It is responsible for providing core APIs, which are required to setup the streaming context, so that the processing of stream of logs (or any data streams) is done more effectively. We have set these two intervals in Scala language as follows:

scalaval WINDOW_LENGTH = new Duration(100)

scalaval SLIDE_INTERVAL = new Duration(20)

It is clear that, at a particular moment, the streams over the recent past 100 ms will be observed for the processing (WINDOW_INTERVAL) and the observation window will move forward after each 20 ms.

Spark Engine Core: Apache Spark core functionalities are provided by Spark Engine core, which is responsible for establishing spark environment generating and transforming of RDDs is considered.

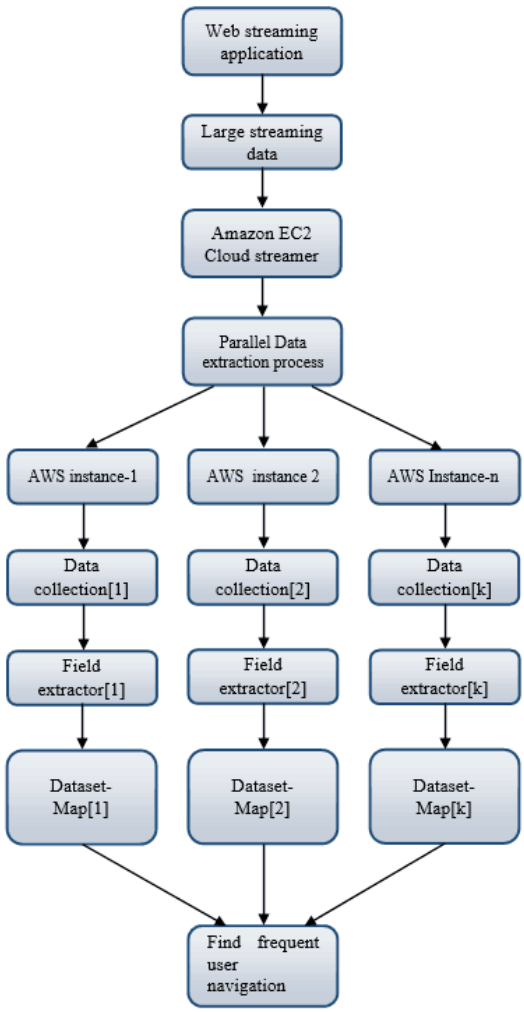

Figure 1. Spark based data collection and user behaviour analysis on large stream data

Spark based data extraction and user behavior analysis is achieved by implementing 3steps or algorithms. First establish the real time streaming server connection for streaming data second Preprocess the large streaming data by creating 3 cluster instances and in third analyzing the user behavior by using Predicting the future navigation pattern mining using historical behaviour and user patterns, and generate the user correlated patterns using min threshold in 1 or more sessions.

Algorithm 1: Real-time Streaming Server Connection for streaming data

Input: Amazon EC2, Streaming log data.

Output: Streaming data fields in cluster instances.

Procedure:

Step 1:

Connection con=Amazon EC2(Instances[]);

Step 2: for each instance in instance list

Do

Setup(Instance[i])

Setup(Spark(Instance[i])

done

Step 3: for each instance in Instancelist

Do

If(instance[i]!==start)

Then

Start instance[i];

Streamer s=getConnection(URL);

if(S.connection==NULL)

return NULL;

Else

Start spark in the instance[i].

LogList[]=getLogData(S);

End if

Done

Step 4: Partition the LogList[] into m number of clusters.

Let logList[]={LogList[0],LogList[1]….LogList[m]};

For each cluster instance

Do

Assign FilteringJob(LogList[i],instance[i]);

Done

In this algorithm, amazon AWS server is used to run the proposed model on the real-time e-commerce streaming dataset. In the step 1, Amazon AWS servers are configured with specified number of cluster instances. In the step 2, for each cluster instance spark framework is installed to perform map reduce operations. In the step 3, each cluster instance is started with the specific data configurations. In this step, streaming log data is captured and stored in the file using the realtime e-commerce websites. In the step 4, each cluster instance is started and data is ready to apply Mapper phase for data filtering. Here, in this step, data is filtered to find the relevant fields and its partitioned data to each cluster instance of Amazon AWS server.

Algorithm 2: FilteringJob(LogList[],instance[])

Input: Streaming Logdata, cluster instances CI.

Ouput: Log streamining fields with filtering

Procedure:

For each cluster instance CI[i]

Do

Fields F[]=LogList[i]

For each field in F[]

Do

If(F[j]==Null &&F[j]==Numerical)

Then

Find Max probability of occurrence in Streaming data at time t.

$$F[j] = \text{Max}\{|\frac{X - F[i]}{\text{Prob}(F[c])}|; \log(|\mu_{F[i]} - \sigma_{F[i]}|).\text{MinMaxNormalization}(F[i])\}c = 1..\#\text{classes}, \quad (1)$$

End if

If(F[j]==Null &&F[j]==Nominal)

Then

$$F[j] = \text{Max Prob}\{\frac{\text{Log}(\text{Prob}(F[i]/F[c]))}{\text{Prob}(F[c])}; \log(F[c]) * \text{Prob}(F[i] \cup F[c])\}c = 1..\#\text{classes}, \quad (2)$$

End if

Done

For each mapper m in M[]

Do

Partition streaming fields to each mapper

M[i]=Partition(F[],k);// K partitions to each mapper

// Finding frequent patterns on each mapper using the FPtree[10] algorithm as initial candidate frequent patterns.

M_Patterns[]=FPtree(M[i],m);//m=1,2,..|M[i]| itemset candidate sets

Done

Done

Streaming data is pre-processed using the algorithm 2. In this algorithm, each cluster instance is processed to extract the streaming web log fields. If the field is numerical type then Eq. (1) is used to find the most possible occurrence of streaming field. Similarly for nominal streaming type, eq (2) is used to find the most possible steaming field using the probabilistic measure. As shown in Figure 1, proposed model is implemented on realtime server with high dimensional streaming e-commerce data. Here, user specified cluster instances are created and configured in Amazon AWS environment to run the proposed model on streaming data. Spark framework is used to partition the data and to integrate the data using the Mapper and Reducer phases. In each cluster instance, a series of operations such as field extraction process, data filtering process and web user navigation behaviour are analyzed using the proposed model. In this model, each mapper is partitioned using the filtered fields of the streaming data. To each partition, frequent patterns on the subset of fields are computed by using the FPtree algorithm. Here, FPtree algorithm is used to construct the initial candidate patterns for the mathematical user behavior analysis in the next model.

Algorithm 3: Web user behavior using feature selection process on streaming data

Input: Cluster instances CI[],Cluster instance filtered data CIFD[], user navigation paths NP, user query.

Ouput: Predicting the user's navigation paths using streaming pattern mining model.

Procedure:

// Predicting the future navigation pattern mining using historical behaviour and user patterns

For each user in the userslist UList[]

Do

For each filtered data CIFD[]

Do

// Predicting the future navigation behaviour using the following prediction formula

Rank[]=

$$\max\{\frac{\prod \text{Prob}(q/\text{CIFD}[j])}{\text{Prob}(q \cap \text{NP}[j]}, \text{linear\_regression}(q, \text{NP}[])\} \quad (3)$$

Mapper(Rank[],CI[]);
Done
Done
//Generate user correlated patterns using η_{min} threshold in 1 or more sessions
FPRules ← ∅;
Find 1-Dimension frequents of user navigations as (It_1) in each session S[]
For each session S[k]
do
for (i = 2, It_{i-1} != ∅ , i + +)
do
(CorrSet$_i$,S[k])←Prob(It_{i-1},It_1)=

$$\frac{Prob(It_{i-1} \cup It_1)}{Correlation(It_{i-1}, It_1)} Prob(\{It_{i-1} \cap It_1\} / NP[]) \quad (4)$$

done
For each K-Set $It_i \in$ *(CorrSet$_i$,S[k])*
do
w_i =ProbVals(Rank[], It_i)
if $w_i \geq \eta_{min}$ then
FPRules[]←Mapper($fps_m \cup \{ It_i , w_i \}$)
end if
done

for each frequent patterns FPRules[]
do
for each cluster instance CI[]
Reduce(FPRules[],Rank[])
Done

Done

In this algorithm, Cluster instances CI[],Cluster instance filtered data CIFD[], user navigation paths NP, user query are taken as input for each spark cluster instance. Initially, this algorithm will predict the future navigation patterns using the ranking measure of eq (3). This ranking measure is used to find the maximization of the navigation pattern related to each user in the Mapper phase. After the future navigation patterns, user based correlated patterns are extracted using the eq (4). Here, each user historical navigation pattern is compared with the streaming data to find the most relevant correlated navigation patterns. These frequent patterns are integrated using the Reducer phase of spark.

4. EXPERIMENTAL RESULTS

In the proposed model, a stream of products information is extracted from the flipkart e-commerce site. Here, the flipkart developer's account is used to extract the products details using the spark instance. Each cluster instance in the cloud is used to find and extract the user's navigation details along with the product features. A sample flipkart dataset is represented in the table 1 in JSON format. Streaming dataset is used to find the essential patterns using the filtering algorithm. Also, a large number of product details are extracted from the flipkart and amazon sites to analyse the user's behaviour towards the navigation paths.

Table 2, describes the filtering of noisy data using proposed filter based model on the large streaming data. In this table, as the size of the data increases proposed approach effectively handles noisy data with less runtime. Table 3, describes the comparison of the proposed model to the existing models in terms of patterns count, true positive rate and error rate. Here, different types of e-commerce products are used to analyse the performance of the proposed model to the existing models. From the table, it is clearly analysed that the proposed model has less error rate and high computational accuracy than the traditional algorithms.

Table 1. Sample E-commerce streaming JSON file

29c8d290caa451f97b1c32df64477a2c 2016-03-25 22:59:23 +0000 http://www.flipkart.com/dilli-bazaaar-bellies-corporate-casuals-casuals/p/itmeh2paagfuhbzh?pid=SHOEH3DZBFR88SCK "dillibazaaar Bellies, Corporate Casuals, Casuals" "[""Footwear >> Women's Footwear >> Ballerinas >>dillibazaaar Bellies, Corporate Casuals, Casuals""]" SHOEH3DZBFR88SCK 699 349 ""http://img6a.flixcart.com/image/shoe/b/p/n/pink-200db202-dilli-bazaaar-10-original-imaeh2zz4x6hnuwf.jpeg"", ""http://img6a.flixcart.com/image/shoe/b/p/n/pink-200db202-dilli-bazaaar-10-original-imaeh2zzxp8s7gru.jpeg"", ""http://img6a.flixcart.com/image/shoe/s/c/k/pink-200db202-dilli-bazaaar-9-original-imaeh2zzv2hzkepv.jpeg"", ""http://img5a.flixcart.com/image/shoe/b/p/n/pink-200db202-dilli-bazaaar-10-original-imaeh2zztckv2tqj.jpeg""]" FALSE "Key Features of dillibazaaar Bellies, Corporate Casuals, Casuals Material: Fabric Occasion: Ethnic, Casual, Party, Formal Color: Pink Heel Height: 0,Specifications of dillibazaaar Bellies, Corporate Casuals, Casuals General Occasion Ethnic, Casual, Party, Formal Ideal For Women Shoe Details Weight 200 g (per single Shoe) - Weight of the product may vary depending on size. Heel Height 0 inch Outer Material Fabric Color Pink" No rating available No rating available dillibazaaar"{""product_specification""=>[{""key""=>""Occasion"",""value""=>""Ethnic, Casual, Party, Formal""}, {""key""=>""Ideal For"", ""value""=>""Women""}, {""key""=>""Weight"", ""value""=>""200 g (per single Shoe) - Weight of the product may vary depending on size.""}, {""key""=>""Heel Height"", ""value""=>""0 inch""}, {""key""=>""Outer Material"", ""value""=>""Fabric""}, {""key""=>""Color"", ""value""=>""Pink""}]}"4044c0ac52c1ee4b28777417651faf42 2016-03-25 22:59:23 +0000 http://www.flipkart.com/alisha-solid-women-s-cyclingshorts/p/itmeh2fvdphhyshh?pid=SRTEH2FVUHAAVH9X Alisha Solid Women's Cycling Shorts "[""Clothing >> Women's Clothing >> Lingerie, Sleep & Swimwear >> Shorts >> Alisha Shorts >> Alisha Solid Women's Cycling Shorts""]"SRTEH2FVUHAAVH9X 1199 479 ""http://img5a.flixcart.com/image/short/5/z/c/altght4p-39-alisha-38-original-imaeh2d5ar6m55zd.jpeg"", ""http://img5a.flixcart.com/image/short/z/h/b/altght-9-alisha-36-original-imaeh2d5khxcdggw.jpeg"",""http://img6a.flixcart.com/image/short/z/h/b/altght-9-alisha-36-original-imaeh2d5yj4cnjtz.jpeg"", ""http://img6a.flixcart.com/image/short/z/h/b/altght-9-alisha-36-original-imaeh2d55eacbgwg.jpeg""

Table 2. Proposed filtering approach onsparsity problem

Streaming Datasize	Filtered Noisy data	Runtime (mins)
#1GB	121MB	4.63
#2GB	653MB	7.13
#5GB	1424MB	9.73
0GB	1964MB	14.26
#15GB	2734MB	17.74

Figure 2, describes the comparison of the proposed model to the existing models in terms of filtered navigation patterns from large candidate sets. Here, different types of e-commerce products are used to analyse the performance of the proposed model to the existing models. From the figure, it is clearly analysed that the proposed model has high filtering capability than the traditional algorithms in order to remove the noise or duplicate patterns.

Table 3. Comparison of proposed model with different models

Datasize=10GB			
Models	**Navigation Patterns**	**True Positive Rate**	**Error Rate**
Rule based pattern Mining	63435	0.864	0.243
Bayesian pattern mining	57343	0.896	0.2053
CorrelationBasedPattern Mining	56343	0.91734	0.1943
RankVoting	51574	0.9274	0.1894
Filtered based FPTree mining	42194	0.9573	0.0827

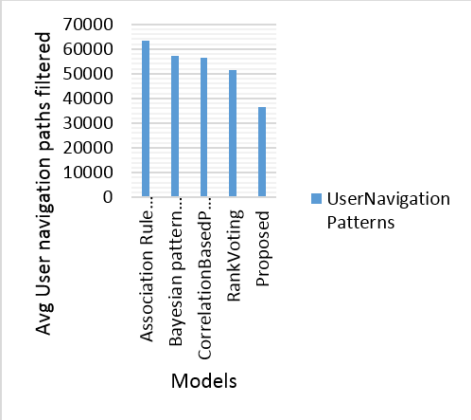

Figure 2. Comparison of the proposed model to the existing models in terms of filtered navigation patterns

Figure 3, describes the comparison of the proposed model to the existing models in terms of true positive rate and error rate. Here, different types of e-commerce products are used to analyse the performance of the proposed model to the existing models. From the figure, it is clearly analysed that the proposed model has less error rate and high computational accuracy than the traditional algorithms.

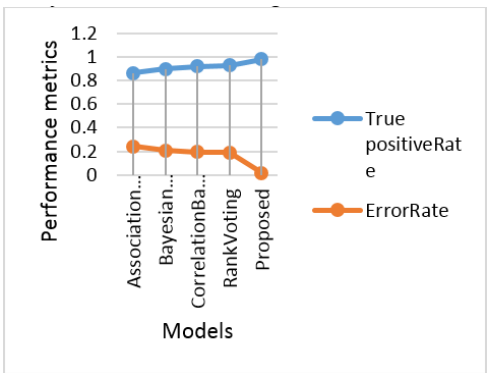

Figure 3. Comparison of the proposed model to the existing models in terms of true positive rate and error rate

5. CONCLUSION

Now a days there is a more value to real time or streaming data than the data stored in darabases. We also discussed the processing of real time and batch data analysis using Spark tool. Analyzation of this real time data is useful for prediction of user behavior in single or multiple websites. In this paper a novel dynamic model is introduced to discover the required knowledge from the large databases. A new filter based user navigation pattern mining model is also designed and implemented on the large online steaming databases. The proposed filtered based frequent pattern mining model efficiently predicts the user navigation patterns with high accuracy and less runtime.

ACKNOWLEDGMENT

This work is supported by University Grants Commission (UGC) under Minor Research Project titled “Development of Mathematical Model for the Prediction of Customer Behavior in Online Databases”.

REFERENCES

[1] Dean J, Ghemawat, S. (2008). Mapreduce: Simpliûed data processing on large clusters. Commun. ACM 51(1): 107-113.

[2] Borthakur D, Gray J, Sarma JS, Muthukkaruppan K, Spiegelberg N, Kuang H, Ranganathan K, Molkov D, Menon A, Rash S, Schmidt R, Aiyer A. (2011). Apache Hadoop goes Realtime at Facebook. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2011), ACM, New York, USA, pp. 1071-1080. https://doi.org/10/2498/cit.1001391

[3] Chen W, Paik I, Li Z. (2017). Cost-aware streaming work flow allocation on geo distributed data centers. IEEE Transactions on Computers 66: 256-271. https://doi.org/10.1109/TC.2016.2595579.

[4] Han, J, HE, Le G, Du J. (2011). Survey on NoSQL database. In: Proceedings of the 6th International Conference on Pervasive Computing and Applications, IEEE, Port Elizabeth, South Africa, pp. 363-366. https://doi.org/10.1109/ICPCA.2011.6106531

[5] Sattler KU, Beier F. (2013). Towards elastic stream processing: Patterns and infrastructure. In: Proceedings of the 1st International Workshop on Big Dynamic Distributed Data (BD3), Riva del Garda, Italy, pp. 49-54.

[6] Dastjerdi AV, Buyya R. (2016). Internet of things: Principles and paradigms. Morgan Kaufmann, Burlington, USA.

[7] Wu E, Diao Y, Rizvi S. (2006). High-performance complex event processing over streams. In: ACM SIGMOD International Conference on Management of Data, SIGMOD 06, ACM, New York, USA, pp. 407-418. https://doi.org/10.1145/1142473.1142520

[8] Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI'12, USENIX Association,

Berkeley, USA, pp. 2-12. https://doi.org/10.1504/IJASM.2015.068610
[9] Zaharia M, Das T, Li H, Hunter T, Shenker S, Stoica I. (2013). Discretized streams: Fault-tolerant streaming computation at scale. In: Proceedings of the 24th ACM Symposium on Operating Systems Principles, SOSP '13, ACM, New York, USA, pp. 423-438. https://doi.org/10.1145/2517349.2522737
[10] Wolfram W, Gessert F, Friedrich S, Ritter N. (2016). Real-time stream processing for Big Data. it-Information Technology 58(4): 186-194. https://doi.org/10.1515/itit-2016-0002
[11] Mavridis EK. (2015). Log file analysis in cloud with apache Hadoop and apache spark. Proceedings of 2nd International Workshop on Sustainable Ultrascale Computing Systems, pp. 51-62. https://doi.org/10.1504/IJICT.2016.079962
[12] Dean J, Ghemawat S. (2004). MapReduce: Simplified Data Processing on Large Clusters. In OSDI'04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation (Berkeley, CA, USA), USENIX Association, p. 10.
[13] Spark Streaming Programming Guide, Spark Streaming - Spark 2.2.1 Documentation. [Online]. Available: http://spark.apache.org/docs/latest/streaming-programming guide.html, accessed on 10 October, 2018.
[14] Patil SD. (2013). Use of web log file for web usage mining. International Journal of Engineering Research & Technology (IJERT) 2(4).
[15] Shahrivari S. (2014). Beyond Batch Processing: Towards Real-Time and Streaming Big Data. Computer 3(4): 117-129. https://doi.org/10.1504/IJIDS.2016.075789
[16] Tyagi NK, Solanki AK, Tyagi S. (2010). An algorithmic approach to data preprocessing in web usage mining. International Journal of Information Technology and Knowledge Management 2(2): 279-283.
[17] Verma V, Verma A, Bhatia S. (2011). Comprehensive analysis of web log files for mining. International Journal of Computer Science Issues (IJCSI) 8(6): 199-202.
[18] Mohamed N, Al-jaroodi J. (2014). Real-time big data analytics: Applications and challenges. International Conference on High Performance Computing & Simulation (HPCS), pp. 305-310. https://doi.org/10.1109/HPCSim.2014.6903700
[19] Nguyen DT, Jung JE. (2016). Real-time event detection for online behavioral analysis of big social data. Future Generation Computer Systems 137-145. https://doi.org/10.1016/j.future.2016.04.012
[20] Cha S, Wachowicz M. (2015). Developing a real-time data analytics framework using Hadoop. 2015 IEEE International Congress on Big Data, pp. 657-660. https://doi.org/10.1504/IJIT.2018.090878
[21] Liu X, Iftikhar N, Xie X. (2014). Survey of real-time processing systems for Big Data. In Proceedings of the 18th International Database Engineering & Applications Symposium, pp. 356-361. https://doi.org/10.1145/2628194.2628251
[22] Patel B, Birla M, Nair U. (2012). Addressing big data problem using Hadoop and map reduce. Nirma University International Conference on Engineering (NUiCONE), pp. 1-5. https://doi.org/10.1109/NUICONE.2012.6493198
[23] Xu D, Wu D, Xu X, Zhu L, Bass L. (2015). Making real time data analytics available as a service. In Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures, QoSA 15: 73-82. https://doi.org/10.1145/3075564.3078884
[24] Deng L, Gao J. (2015). An advertising analytics framework using social network big data. Institute of Electrical and Electronics Engineers 470-479. https://doi.org/10.1109/ICIST.2015.7289018