Classification Of Massive Data Streams Using Naïve Bayes

Mrs.Shiela David[1], K.Ranjithkumar[2], Saurabh Rao[3], Santhosh Baradwaj[4]
D. Sudhakar[5]

[1]**Assistant Professor,**
[1][2][3][4][5]**SRM Institute of Science & Technology,Ramapuram Campus,Chennai,India.**
shielarani23@gmail.com, kranjith004@yahoo.com, saurabh.rao321@gmail.com
santhoshbaradwajvr306@gmail.com, sudhakarmurthy555@gmail.com

Abstract: A coexisting challenge in Machine Learning includes mining of large complex data and aggregating the knowledge. With the rapid generation of data from different sources such as sensor data, Internet of Things data, Networking data, Market Share data and so on we have a massive amount of data generated that are not analyzed to obtain the maximum benefits. These massive data must be explored efficiently and converted into valuable knowledge which could be used by respective fields to enhance their performance or predict a non-existing event. Therefore this paper implements highly efficient and popular algorithm "Naïve Bayes Algorithm" on huge complex data to acquire knowledge. An addition reduction technique is used to eliminate similar data which in turn reduces the computation time, demands less memory space comparatively and enhances the performance of Naïve Bayes Algorithm. HDFS provides a platform with clusters of distributed systems which provides the functionality of storage and the processing. The unstructured or semi-structured data are reconstructed to required file format by converting into CSV, followed by converting into ARFF. Weka tool is a machine learning tool which is used to apply the proposed algorithm on the massive data streams. The functionality of the proposed solution is proven by an experimental study conducted on a set of real-life massive data streams and indicates that we are able to provide the highly efficient Naive Bayes Algorithm solution for huge data streams.

Keywords*—*Naïve Bayes, Reduction, HDFS(Hadoop Distributed File Systems), Clusters of Distributed systems, Data Streams, Weka Tool, CSV(Comma-Separated Values) ARFF(Attribute Relation File Format)

I.INTRODUCTION

In the current generation, almost every human activities involve the use of a computerized system, for instance, our smartwatch sensors, surveillance devices, Billing systems, Social Media, Share Market, Weather Forecast, IT industry, Medical devices, NASA and the list goes on. Due to the usage of a system, there arises a scenario to store the generating data from the system for further processing. In most of the cases, the data generated from the system are kept inactive. It is estimated that 2.5 quintillion bytes of data are generated every day, it is a massive amount of data as 90% of the data in the world today has been created in the last two years alone. Also by considering Moore's law, it states that processing capacity doubles every 18 months, while disk storage capacity doubles every 9 months which indicates that the data productivity by the system is increasing at a rapid speed. By negotiating all the above facts, the future of us has to deal with processing of huge data sets of different fields thus needs an effective method to analyze and process the future demands.

The challenge that we face today is that there are not enough initiative to analyze and grasp the knowledge from the generated data with which the performance of the process can be increased or can predict an outcome of an event which is still a hypothesis. Thus we have to introduce and implement statistical and machine learning algorithms to overcome the above factor.

The main aim of classification of the data stream is to analyze and mine the knowledge from the massive data stream and predict the outcome of a non-existent input stream which is near to hypothesis. Initially, all the data streams obtained from the client are stored in HDFS (Hadoop Distributive File System). Here the data streams are bisected equally with respect to the cluster of distributed systems available in the HDFS platform. Once the data streams are stored, they are converted into Comma-Separated Systems format followed by converting them into ARFF file format. A machine learning tool is used to classify each attribute and place it in the respective attribute type. Then by using a reduction technique, the duplicate data are eliminated which intern provide only unique data stream. Now by using Naïve Bayes Algorithm, the probability of each attribute in the data stream is estimated with respect to its outcome. With the acquired probabilities of the individual attribute, the system is able to predict the non-existent attribute set. The result is provided in the form of a ratio that provides the outcome of the non-existent stream that was given by the client.

Organization of the paper

Initially, the Literature survey of the paper is discussed followed by the proposed system with its advantages and workflow diagram. Then the analysis and calculation of the paper are discussed. The reference paper used for formulating this paper is mentioned at the end.

II.LITERATURE SURVEY

This section will indicate all the imported literature survey and analysis on mining massive and streaming data with special focus put on Naïve Bayes algorithm classification approach.

Hadoop and Map Reduce

Hadoop is an open source framework which runs on HDFS (Hadoop Distributed File System), its functionality is to store file system that is of huge data set in a cluster of inexpensive hardware. It provides stream access pattern by which the data stored can only be read and cannot be edited. The default memory block size of HDFS is 64MB with which it makes use of unused memory block as the data streams are stored consecutively one after the other [3]. HDFS provides five services (i.e) Name Node, Secondary Node, Job Tracker, Data Node and Task Node. By using these services, the huge data can be stored and processed in the cluster of computer simultaneously [4].

Map Reduce is a concept of gathering all the output file together from each node of a cluster. Initially, the huge data streams undergo input split where the data are separated in the computer of the cluster. All the individual systems in the cluster process the data simultaneously and come up with its output files independently. The function of the map-reduce is to gather all the files from the individual systems and deliver a single output. [5]

Weka Tool

Weka Tool is a machine learning tool which accepts ARFF (Attribute Relation File Format) as input. Initially, the attribute type and the attributes are provided to the Weka Tool from the client. After which, with respect to the attribute type and attributes given, the tool reads each line of the data stream and looks for the attributes. When it detects an attribute it places the attribute in its respective attribute class of ARFF table. By the above process, all the attributes are placed in the respective attribute class and provide a complete ARFF for unstructured data provided [17].

K-Nearest Neighbour Algorithm

Nearest Neighbour Algorithm is a simple algorithm used in data mining. The role of this algorithm is to read the data and sort them with respect to given condition. Thus when a non-existed input data is provided it counts the number of nearest distance record 'K' to the input provided. With respect to the nearest data to the input, the output is computed. This algorithm is suitable only for small data sets as it provides noisy data when it is implemented in a huge data set. The K value is given by the user which is the number of sample existing record that should be taken into consideration [2].

The below shown diagrams are simple representations of K-NN algorithm. The graph's x-axis denotes the temperature of the human body and y-axis denotes human eye colour. Initially the existing data are plotted in the graph as Disease A and Disease B by considering the disease characteristics. When an unknown disease x is to be determined, the algorithm compares the existing plot with respect to nearest distance records and predicts an output.

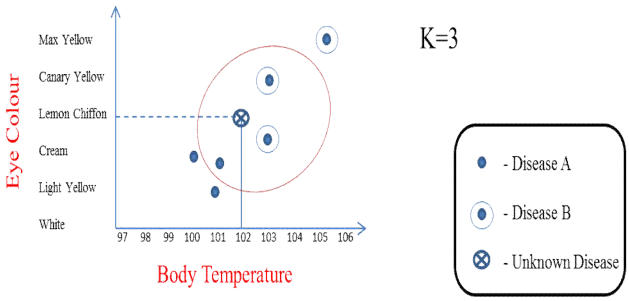

Fig1: K-NN Output when K=3

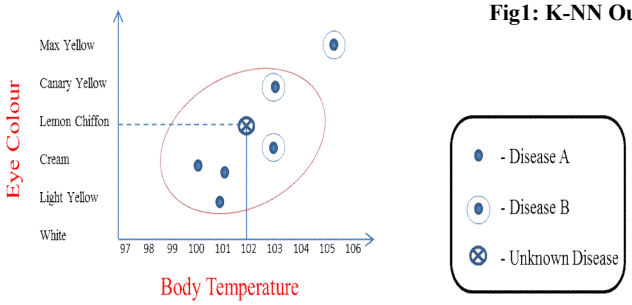

Fig2: K-NN Output when K=5

Disadvantages of the existing system:

1) The accuracy of the K-Nearest Neighbor algorithm is too feeble (i.e) instead of acquiring knowledge from data it simply computes the distance and sort the data. With respect to the nearest distance to the target the output is generated which is slow due to the large number of data provided.

2) In KNN algorithm, the change in the value of k will provide inconsistent predictions for the same data. Also Large data usage will lead to generation of noisy data.

3) The large data consist of many duplicate data which will affect the performance of the algorithm, demands high memory space and also seize a lot time for processing.

III. PROPOSED SYSTEM

In our proposed system, initially the massive data provided by the user are stored in Hadoop Distributed File System. The HDFS provides a platform where the huge data sets are divided with respect to the cluster of distributed systems available and stored in it. The usage of HDFS as improved to overcome data loss factor as it stores each data thrice to prevent data loss. Also, it delivers its service in the form of stream access pattern where the data can only be read.

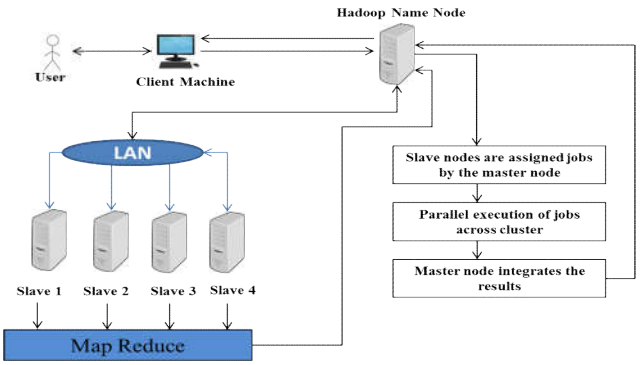

Fig3: Hadoop Distributed File System Architecture

Once the data are stored in HDFS, they are converted into Comma-Separated Systems format followed by converting them into ARFF file format which is acceptable in the proposed machine learning tool. The Machine Learning tool utilized in this paper is Weka Tool. The functionality of this tool is to create an attribute class for each attribute type as instructed by the user. Once the attribute class is built, the tool scans each stream of data to classify its attribute type and drop it into the respective attribute class type, eventually converting all the ARFF file into Attribute Relation Table which will be compatible to apply the algorithm over it.

Now the Reduction technique is implemented over structured data provided by the machine learning tool. The functionality of this technique is to eliminate the duplicate data. This technique makes the data to undergo two steps (i.e) sorting and aggregation. The sorting step arranges the attributes in numerical and alphabetical order. Then the aggregation step looks for any duplicate entries and eliminates it.

After performing all the above procedures, Naïve Bayes algorithm estimate the probability of each and every individual attribute in the data set. With the estimated probabilities an output of a non-existing input data set can be estimated by multiplying all the probabilities of individual attribute of the input data set. Thus obtaining an accurate ratio of the event outcome.

Advantages of the proposed system:

1) The accuracy of the Naïve Bayes algorithm is much more than the existing system. It acquires knowledge from data in the form of probabilities which intern provides the output with high accuracy.

2) Large data streams are high acceptable by the Naïve Bayes algorithm because it does not sort any data. Each and every attribute and its outcome is determined individually which in turn does not produce any noisy data.

3) In the proposed system before applying Naïve algorithm on the large data stream, a reduction technique is used which eliminates the duplicate data and provides data which unique for further process.

IV.ALGORITHM

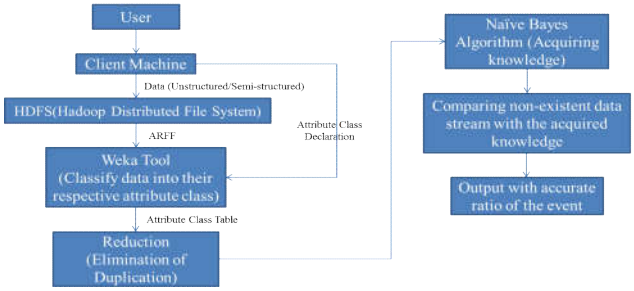

Fig4: Overview of Proposed System

The proposed system undergoes through seven important steps which are shown in the above data flow diagram are listed below:-

Step 1: The user uploads the huge amount of unstructured and semi-structured data into the client machine.

Step 2: The client machine stores the received data in the HDFS (Hadoop Distribute File System).

Step 3: The format of the unstructured data is converted into to Comma-Separated Systems format followed by converting them again into ARFF file format.

Step 4: Using the service of the HDFS, the Weka tool mines each attribute and creates an attribute relation table using Job Tracker and Task Node services.

Step 5: Now a Reduction technique is applied to eliminate the duplication in the data streams.

Step 6: Naive Bayes algorithm is implemented on the reduced attribute relation table by which the algorithm acquires knowledge from the data streams in the form of probabilities.

Step 7: Now a non-existing input data streams is taken and compared with the acquired knowledge to predict its output.

Step 8: An output is generated with the accurate ratio of the event outcome.

V.RESULT AND ANALYSIS

Thus the utilisation of the proposed system will provides highly accurate results while comparing the existing system.

The reduction technique helps to provide unique data set for applying the algorithm. The Naïve Bayes Algorithm consider each occurrence of an attribute in the existing data stream and delivers the output of non-existing input by studying the probabilities of existing individual attribute.

VI.ACCURACY AND PROCESSING

Accuracy Graph

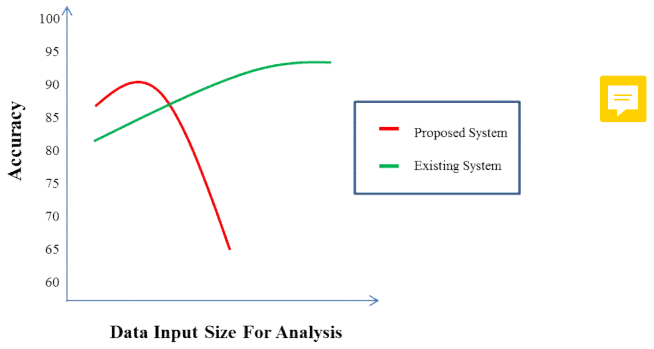

Fig5: Representation of Accuracy

While considering the above graph when the input data size increases the existing graph's accuracy proportionally decreases whereas the proposed system is stable while dealing with huge data streams. This is because the proposed system does not produce any noise with respect to increase in data input.

Processing Time

Due to the usage of Hadoop framework the proposed system takes much time for less input data for processing while comparing with the existing system. But when the Data stream size increases enormously the processing time taken by the proposed system is less than the existing system.

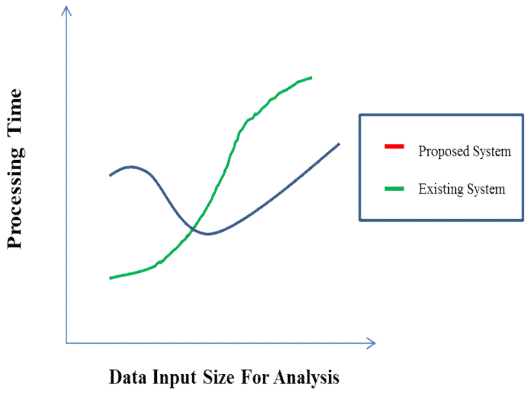

Fig6: Representation of Processing Time

VII.HARDWARE REQUIREMENTS

The implementation of the above proposed system requires a cluster of commodity hardware to provide storage and process the high-speed data stream. When a huge structured or unstructured or semi-structured data is uploaded to the main computer, the files are equally divided and sent to the sub-computers with respect to its memory capacity.
Cluster of computer:

A computer cluster consists of a set of loosely connected computers of the similar type that are interlinked by coaxial, optical fiber, or twisted pair cable network connection. Through fast LANs, the components of a cluster are generally connected to each other, with each node running its own operating system.

Minimum Cluster Node Requirements:
1) 1 GB RAM
The node computer requires a minimum of 1 gigabytes of Random Access Memory i.e volatile memory to compute the required application and the data by the OS.
2) 80 GB Hard Disk
80 GigaByte is the minimum non-volatile memory required by each node. The memory block size is partitioned by 64MB or 128MB as default by Hadoop Distributive File system to abort the unused memory space in each block.
3) Intel Processor
Intel Processor helps to process the information or instruction provided by the computer and the software. Core I3, I5 & I7 version are compatible with the proposed system.

LAN:
All the nodes are interconnected using the Local Area Network. The LAN can consist of N number of nodes which needs a repeater to amplify the signals to the very end of the LAN cable.

VIII.SOFTWARE REQUIREMENTS

1) Windows 7 Ultimate 32-Bit
It is a basic user-friendly operating system which is widely used in most of the current personal computers and Laptop. Hadoop Distribution File system is capable to perform in the windows platform version window 7 ultimate 32 bit and higher.
2) JDK 1.7
Java Software Developing Kit is used to perform the necessary algorithm over the assigned data. It provides an enormous amount of tools for developing, debugging and monitoring etc. The main reason to opt for JDK is that it is platform independent, robust and secure.

3) Weka Tool

Weka tool consists of varies machine learning algorithms which can be directly applied to the huge data sets through the java commands. With this tool, we categorize the attributes in the respective attribute type and create an attribute relation table.

IX.CONCLUSION

Thus the classification of Data streams using Naïve Bayes algorithm helps the client by acquiring knowledge from the massive data provided and predict the outcome of a hypothesis data input without generating any noisy data intern achieves high accuracy.

Due to the usage of reduction technique, the duplicate data are eliminated and only unique data streams are considered which in turn demands less memory space and less processing time. The implementation of the Naïve Bayes algorithm provides high accuracy outcome of the event which makes the proposed system to deliver elevated performance than other existing systems.

X.FUTURE SCOPE

The proposed system is just a basic initiation taken to analyze and study data as analysis of huge data streams is at the beginning stage for the current generation. The future scope of this paper involves implementing many different statistical and relative algorithm on the data streams to increase the better study and analysis of the data which intern enhance the accuracy of the computer to acquire knowledge deeply and predict the outcome of hypothesis events.

REFERENCES

[1] Sergio Ramírez-Gallego; Bartosz Krawczyk; Salvador García; Michał Woźniak; José Manuel Benítez; Francisco Herrera "Nearest Neighbor Classification for High-Speed Big Data Streams Using Spark" IEEE Transaction Oct 2017.

[2] J. Maillo, S. Ramirez, I. Triguero, and F. Herrera, NN-IS: An iterative spark-based design of the k-nearest neighbors classifier for big data, Knowl. Based Syst., vol. 117, pp. 15, Feb. 2017.

[3] Apache Spark: Lightning-Fast Cluster Computing (2017).

[4] Apache Hadoop Project (2017).

[5] Y. Xun, J. Zhang, and X. Qin, Parallel mining of frequent item sets using MapReduce, IEEE Trans. Syst., Man, Cybern., Syst., vol. 46, no. 3, pp. 25, Mar. 2016.

[6] W.-P. Ding, C.-T. Lin, M. Prasad, S.-B. Chen, and Z.-J. Guan, Attribute equilibrium dominance reduction accelerator (DCCAEDR) based on distributed coevolutionary cloud and its application in medical records, IEEE Trans. Syst., Man, Cybern., Syst., vol. 46, no. 3, pp. 384-400, Mar. 2016.

[7] Katrina Sin, Loganathan Muthu, "Application of Big Data in Education Data Mining and Learning Analytics - A Literature Review", ICTACT Journal on Soft Computing ISSN, vol. 5, no. 4, pp. 2229-6956, July 2015.

[8] A. R. Mahmood et al., Tornado: A distributed spatio-textual stream processing system, in Proc. 41st Int. Conf. Very Large Data Bases, vol. 8. pp. 2020-2023, 2015.

[9]S. Bhosale Harshawardhan, P. Gadekar Devendra, "A Review Paper on Big Data and Hadoop" in IJSRP ISSN, vol. 4, no. 10, pp. 2250-3153, Oct 2014.

[10] Mimran and A. Even, Data stream mining with multiple sliding windows for continuous prediction, in Proc. 22nd Eur. Conf. Inf. Syst. (ECIS), Tel Aviv, Israel, 2014.

[11]Lorch et al, 2013, J. Lorch, B. Parno, J. Mickens, M. Raykova, and J. Schiffman, Shoroud: Ensuring Private Access to Large-Scale Data in the Data Center, In: Proc. of the 11th USENIX Conference on File and Storage Technologies (FAST'13), San Jose, CA, 2013.

[12] Silva et al. 2012, Alzennyr da Silva, Raja Chiky, Georges Hébrail, A clustering approach for sampling data streams in sensor networks, Knowledge and Information Systems, July 2012, Volume 32, Issue 1, pp 1-23.

[13] E. Birney, "The Making of ENCODE: Lessons for Big-Data Projects," Nature, vol. 489, pp. 49- 51, 2012.

[14] Labrinidis and Jagadish 2012, A. Labrinidis and H. Jagadish, Challenges and Opportunities with Big Data, In Proc. of the VLDB Endowment, 2012.

[15] T. White, Hadoop, the Definitive Guide. Sebastopol, CA, USA: O'Reilly Media, 2012.
[16] H. He, S. Chen, K. Li, and X. Xu, Incremental learning from stream data, IEEE Trans. Neural Netw., vol. 22, no. 12, Dec. 2011.
[17] Arff convertor tool for WEKA data mining software, R. Robu ; V. Stoicu-Tivadar, June 2010.
[18] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Google Inc, 2010.
[19] Xin YueYang, Zhen Liu, Yan Fu, "MapReduce as a Programming Model for Association Rules Algorithm on Hadoop", Information Sciences and Interaction Sciences (ICIS) 2010 3rd International Conference on, pp. 99-102, 2010.
[20] Naïve Bayes Text Classifier by Aiyi Zhang ; Di Li IEEE Nov 2007.