Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.11 No.1, 2022, pp.1-4
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajcst-2022.11.1.3071

Predictive Analysis on Sensor Data Using Distributed Machine Learning

S. Shargunam and G. Rajakumar
Department of Electronics and Communication Engineering, Francis Xavier Engineering College, Tamil Nadu, India
E-mail: shargunamguna@gmail.com, gmanly12@gmail.com

(Received 8 December 2021; Revised 18 December 2021; Accepted 5 January 2022; Available online 12 January 2022)

Abstract - **The science of getting computers to act without being explicitly programmed is known as machine learning. Machine learning is comparable to data mining in terms of how it works. Both systems sift through data in order to find patterns. Machine learning, on the other hand, instead of extracting data for human comprehension as in data mining applications, uses that data to increase the program's own understanding. Data patterns are detected by machine learning programmes, which then alter Programme behaviours accordingly. This research is based on real-world data obtained from sensors in an oil and Gas Corporation that monitor drilling procedures and equipment. The sensor data is streamed in at a one-second period, resulting in 86400 rows of data per day. Ox data's H2O has chosen for this problem after researching state-of-the-art Big Data analytics tools such as Mahout, RHadoop, and Spark because of its rapid in-memory processing, robust machine learning engine, and ease of use. Missed values can be estimated using accurate predictive analytics of massive sensor data, or wrong readings can be replaced owing to malfunctioning sensors or a broken communication line. It can also be used to predict circumstances to aid in various decision-making processes, such as maintenance planning and operation. In this project, sensor data has been evaluated and anticipate output using the H2O tool. The machine learning techniques has been employed in distributed systems, such as connecting five nodes to accomplish parallel processing.**

Keywords: **Hadoop, Distributed Systems, Machine Learning, Sensors**

I. INTRODUCTION

Machine learning is a branch of computer science that arose from artificial intelligence's pattern recognition and computational learning theory. Machine learning entails the research and development of algorithms that can learn from and predict data. Rather of using strictly static computer instructions, these algorithms develop a model from sample inputs in order to make data-driven predictions or judgments. Computational statistics, which tries to build algorithms for implementing statistical approaches on computers, is strongly related to machine learning. It is closely linked to mathematical optimization, which provides the discipline with tools, theory, and application domains. Machine learning is used in a variety of computing jobs when explicit algorithms are impossible to design and programme. The primary goal of this project is to forecast the outcome of sensor data that are to be used in oil and gas firms in the event of a failure. Instead of employing a traditional machine learning technique, a distributed machine learning technique was adopted. The fundamental advantage of distributed machine learning over traditional machine learning is that it uses numerous nodes to process data in parallel, which saves time and improves performance and scalability. H2O tool has been chosen for this parallel processing since it is the most effective analysis tool when compared to other tools like R-Hadoop, Mahout, Spark, and others. In this project, gas sensor data has been collected and use the H2O tool to perform parallel processing on distributed multiple nodes, evaluate performance, and predict the output or future result of the sensor data set. For predictive analysis of sensor data collected from gas sensors, distributed machine learning approach was adopted rather than traditional machine learning. Existing machine learning is more efficient in terms of performance and scalability as a result of this.

II. RELATED WORKS

Distributed data on a network of arbitrarily connected computers without data exchange was proposed by Leonidas Georgopoulos *et al.,* [5]. For the goal of proving the theoretical conclusions, a distributed update equation of a feed-forward neural network with back-propagation was created. In the general scenario where the learning algorithm is a contraction, proof of convergence of the distributed learning process. It's possible that the total effort result higher. Computing time is proportional to the number of machines in the network, and local datasets are computed in parallel.

Simone Scardapane *et al.,* [6] proposed developing distributed learning algorithms for random vector functional –link (rvlf) networks with distributed training data and a decentralized information structure. Decentralized average consensus (DAC) and alternate direction method of multipliers (ADMM) techniques are used to propose two algorithms. During the learning process, these algorithms work in a fully distributed manner and do not require coordination from a central agent. The starting weight of the input layer is known by all stations, while the output weights of RVFL networks can only be communicated across surrounding nodes via communication channels. Local datasets are also strictly prohibited. These proposed techniques are tested on five different datasets. Further research in this field could include algorithmic advances as well as applications where centralized solutions aren't possible. Furthermore, the current technique is intended to be extended to online learning for RVFL and to the situation of recurrent hidden layers (e.g. Echo state networks).

Habib mostafaei *et al.,* [7] advocated that the study on barrier coverage include energy efficiency as a goal. In this study, cost can refer to any performance metric and is typically described as any resource consumed by the sensor barrier. The stochastic barrier coverage problem entails determining the smallest number of sensor nodes required to build a sensor barrier path. The problem of barrier coverage is first studied using a stochastic coverage graph. Then, to discover a near-optimal solution to the stochastic barrier coverage problem, a distributed learning automata-based method is given. Furthermore, simulation studies should be carried out, with the results demonstrating that this approach can effectively increase the number of barriers paths in a wireless sensor network across a wide range of deployment nodes.

For Chinese entity relation extraction, Lishuang Li *et al.,* [8] suggested a distributed meta-learning method that includes the distributed system and the meta-learning strategy. At the most fundamental level of meta-learning, a learner for each connection type has been created, and the basic learners differ from one another due to distinct feature sets. Experiments on Automatic Content Extraction are being conducted. The F-score of our distributed meta-learning system is 69.81 percent, which is greater than that of the baseline (the approach based on Support Vector Machine (SVM) utilizing composite kernel) by 1.31 percent and surpasses the state-of-the-art systems.

For Chinese entity relation extraction, a distributed meta-learning technique that integrates a distributed system and a meta-learning strategy has been developed. At the most fundamental level of meta-learning, a learner for each connection type has been created, and the basic learners differ from one another due to distinct feature sets. Then, in order to boost performance, communication among these fundamental learners is established. The relation subtypes are retrieved, and our system are subjected to rules-based post-processing. The existing result has changed to increase precision based on the consistency between relation types and relation subtypes. Furthermore, because several semi-supervised methods have been extensively explored and proven to perform better which supplement our system with a large size unlabeled corpus.

The effects of process-level information on machine learning prediction outcomes are defined, and the effects of the type of machine learning algorithm utilised on prediction results are established, according to Alexander J Stimpson *et al.,* [9]. For sophisticated algorithms, process level information is employed to provide helpful prediction features in a more precise manner. Regression and classification analyses on the dataset are used to determine the usefulness of process-level information in machine learning prediction models. Intervention approaches must further translate complex data sets into prediction findings.

III. PROPOSED METHODOLOGY

A. Problem Description

To solve real-time problems with big data analytics and to apply distributed machine learning techniques in an efficient manner to improve speed and scalability. This research is based on real-world data obtained from sensors in an oil and gas corporation that monitor drilling procedures and equipment. The sensor data is streamed in at a one-second period, resulting in 86400 rows of data per day.

We chose Ox data's H2O for this problem after researching state-of-the-art Big Data analytics tools such as Mahout, R-Hadoop, and Spark has been chosen because of its rapid in-memory processing, robust machine learning engine, and ease of use. Missed values can be estimated using accurate predictive analytics of massive sensor data, or wrong readings can be replaced owing to malfunctioning sensors or a broken communication line. It can also be used to anticipate situations that aid in various decision-making processes, such as maintenance and operation planning.

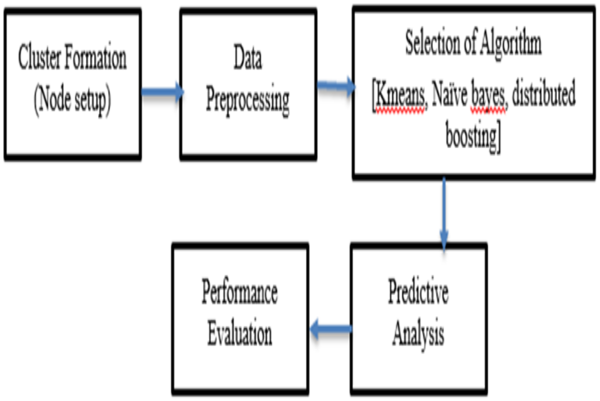

Fig. 1 Flowchart of the system

B. Cluster Formation

In this Cluster Formation module, n-nodes are to be created in order to create distributed machines. This notion is built on the master-master paradigm, and all nodes have master rights. Here, a wired network connects all of the nodes. The data has gathered from the UCI repository, which included Sensor gas data with 20000 attributes and 1 lakh rows. The class label, weight, mass, velocity, and gas type are all included in our dataset.

C. Parsing

The parallel processing of the dataset has been collected. By using this parsing technique, numerous nodes are specified and performing data preprocessing. This preprocessing is done on the dispersed nodes to improve the system's performance and efficiency. In the H2O tool, the number of nodes are specified and memory space has been allotted for each node before parsing.

D. Algorithm Selection

After preprocessing the dataset in various nodes, we'll choose an algorithm for predicting and evaluating the dataset's performance. Default algorithms in the H2O tool include the nave-baiyes algorithm, k-means algorithm, and distributed boosting algorithm. To move forward, one of the algorithms has been executed.

E. KMEANS Algorithm

In this problem, the k-means algorithm has been implemented, which provides a better and more efficient result than the other two algorithms. The k-nearest neighbour classifier, a common machine learning technique for classification that is sometimes confused with k-means due to the k in the name, has a tenuous relationship with this algorithm. To categorise fresh data into existing clusters, the 1-nearest neighbour classifier can be applied to the cluster centers acquired by k-means. This is known as the Rocchio algorithm or nearest centroid classifier.

F. Predictive Analysis

By default, the H2O tool has a standardized value. The datasets provided as input has be broken down into smaller chunks and processed. The outcome presented as a checksum value. The tool then compares the check sum value and standardized value for given data, selecting the dataset with the largest difference as the malfunctioning dataset, allowing us to determine where the error occurred and anticipate future output.

G. Tool Description

H2O makes it simple for anyone to tackle today's most difficult business problems using arithmetic and predictive analytics. It includes intelligently unique characteristics not available in other machine learning platforms, such as: Best of Breed Open Source Technology - Enjoy the freedom that comes with using Open Source technology to power big data science. H2O uses the most widely used Open Source tools, including as Apache TM Hadoop and Spark TM, to provide customers with the flexibility they need to address their most difficult data challenges.

Easy-to-use WebUI and Recognizable User Interfaces - Use H2O's easy Web-based user interface or familiar programming environments like R, Java, Scala, Python, JSON, and our powerful APIs to rapidly set up. Support for all common database and file types that is data agnostic - Explore and model huge data from Microsoft Excel, R Studio, Tableau, and other applications. Data from HDFS, S3, SQL, and NoSQL data sources can be accessed. It can be Installed and used from any location. Massively Scalable Big Data Analysis - With H2O's quick in-memory distributed parallel processing, a model can be trained on whole data sets, not just small samples, and iterate and create models in real-time. Real-time Data Scoring - in any setting, the Nanofast Scoring Engine has used to score data against models for precise predictions in nanoseconds.

Enjoy 10X faster scoring and predictions than the market's next closest technology. In our case, if the sensor equipment is broken or malfunctions, it is unable to provide the desired result. In that situation, this tool has been used to anticipate the precise output for sensor data; however, the programme with the accessible data set as input are provided. For this, various algorithms such as k-means, distributed boosting algorithm has been utilized.

H. Module

A feature named “Water Meter” in the H2O tool clearly displays the list of linked system's load, user time, number of cores, and system time, among other things. This feature isn't available on the Windows platform. As a result, the data has implemented in Linux. H2O requires four times the amount of RAM as the size of the data set to be analyzed. As a result, each node some memory at the start has been received.

Each node's allocated RAM has been be shared among the linked nodes within a cluster via IP address and port number. The collected data set are loaded, allowing each attribute of the data set to be subjected to feature selection. After that, information about the selected qualities, such as range, mean, NaN, are displayed. The computers that must be connected as a cluster employ a text file that specifies the IP and port number of the nodes that must be connected. All of these systems can share their valid resources through this text file.

IV. EXPERIMENTAL RESULTS

To improve prediction accuracy, the given data set has randomly splitted into frames such as test and train. The resulting checksum value has been compared to the same data set's standard checksum value.

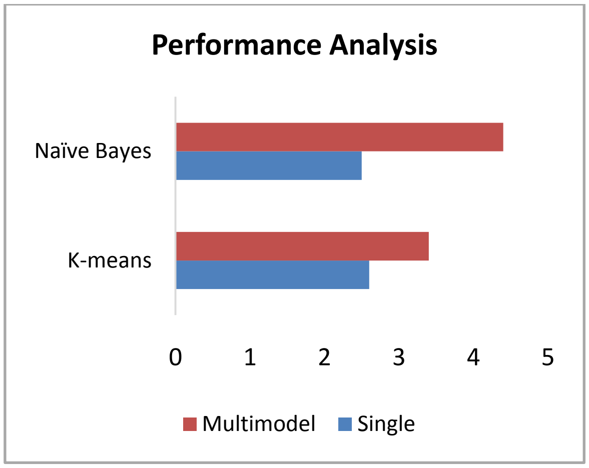

Fig. 2 Performance Analysis

If the deviation is within the permissible range, the result is true (i.e., the sensor is functioning properly); otherwise, it needs to be replaced or fixed. For both single node and multi node predictions, the time it takes to complete the forecast has been recorded. As a result, parallel processing enhances performance over single-node processing. Multi node also helps to lessen system load by allowing background jobs to run. The difference has been displayed as a graph which shown in the Figure 2.

V. CONCLUSION

After analyzing the data using various algorithms predicted checksum value has been compared for the given data set to the standard checksum value for the same dataset without any sensor malfunctioning; if the difference is within the acceptable range, it predicts that there is no malfunction and that the sensor does not need to be replaced; if the difference is not within the acceptable range, it predicts that the sensor needs to be replaced. After comparing regular machine technique and distributed machine learning technique, we choose the distributed machine learning system, a better and more efficient performance for forecasting sensor data analysis results has been provided.

REFERENCES

[1] J. Stimpson, Alexander and L. Mary, Cummings, "Assessing intervention timing in computer-based education using machine learning algorithms," *IEEE Access 2*, pp. 78-87, 2014.

[2] L. Clifton, D. A. Clifton, M. A. Pimentel, P. J. Watkinson, and L. Tarassenko, "Predictive monitoring of mobile patients by combining clinical observations with data from wearable sensors," *IEEE Journal of Biomedical and Health Informatics,* Vol. 18, No. 3, pp. 722-73, 2013.

[3] A. Rahman, D. V. Smith and G. Timms, "A novel machine learning approach toward quality assessment of sensor data," *IEEE Sensors Journal*, Vol. 14, No.4, pp.1035-1047, 2013.

[4] H. Mostafaei, M. Esnaashari, and M. R. Meybodi, "A coverage monitoring algorithm based on learning automata for wireless sensor networks," arXiv preprint arXiv: 1409.1515, 2014.

[5] L. Georgopoulos, and M. Hasler, "Distributed machine learning in networks by consensus," *Neuro Computing,* Vol. 124, pp. 2-12, 2014.

[6] S. Scardapane, D. Wang, M. Panella, and A. Uncini, "Distributed learning for random vector functional-link networks," *Information Sciences*, Vol. 301, pp. 271-284, 2015.

[7] H. Mostafaei, "Stochastic barrier coverage in wireless sensor networks based on distributed learning automata," *Computer Communications*, Vol. 55, pp. 51-61, 2015.

[8] L. Li, J. Zhang, L. Jin, R. Guo, and D. Huang, "A distributed meta-learning system for Chinese entity relation extraction," *Neuro Computin*g, Vol. 149, pp. 1135-1142, 2015.

[9] X. Bi, X. Zhao, G. Wang, P. Zhang, and C. Wang, "Distributed extreme learning machine with kernels based on map reduce," *Neuro Computing*, Vol. 149, pp. 456-463, 2015.