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

Revue d'Intelligence Artificielle
Vol. 36, No. 4, August, 2022, pp. 553-560
Journal homepage: http://iieta.org/journals/ria

Faulty Node Detection in HDFS Using Machine Learning Techniques

Reshma S. Gaykar[1*], Velu Khanaa[1], Shashank D. Joshi[2]

[1] Bharath Institute of Higher Education and Research, Chennai 600073, India
[2] College of Engineering, Bharati Vidyapeeth (Deemed to be University), Pune 411043, India

Corresponding Author Email: reshma.gaykar@gmail.com

https://doi.org/10.18280/ria.360406

Received: 18 July 2022
Accepted: 19 August 2022

Keywords:
HDFS, data node, faulty nodes, machine learning, distributed systems, master node, map reduce, virtual machines

ABSTRACT

The design of Hadoop has ability to elimination of fault tolerance, which consists of rescheduling the task on the defective nodes to run on other devices in the system. However, this strategy is ineffective if an error arises after most of the task has been completed. As a result, it is essential to make an early detection of the problem at the node to ensure that the resumption of the work will not result in a significant loss of both time and productivity. The ability to predict these problems provides us with the required time to move the workload onto different nodes, which helps to avoid data loss or processing time. In this paper, we propose an identification of faulty nodes from a large Hadoop distributed environment using machine learning techniques. Initially, we deployed one controller node and numerous data nodes as virtual machines in a distributed manner. The execution performs when the end-user submits a specific job to the controller node. The master node is the middleware controller that continuously communicates with other data nodes and assigns a task to each data node accordingly. However, this conventional process of HDFS that can generate data leakage or high computation whenever the specific node is heated or straggler. In our approach, we initially collect the log history of each data node and apply some statistical and a few machine learning algorithms to identify nodes' status. According to the achieved outcome of each node, we can decide to eliminate the specific node for task execution. We applied five machine learning algorithms in the extensive experimental analysis, including a Support Vector Machine (SVM). The SVM obtains 96.7% higher accuracy over the conventional machine learning classifiers for the entire execution.

1. INTRODUCTION

High resource availability in virtualization allows for the provision of continuous cloud services, notwithstanding the possibility that one or more elements may fail for several reasons. The variety of cloud designs and the huge scale at which they operate have combined to make them more complicated than the infrastructures of conventional distributed applications. Therefore, new contemporary applications for the Internet of Things and the cloud, such as smart cities and healthcare, need new design frameworks that ensure high stability and availability. Customers and cloud service providers are worried about cloud services' availability and dependability. The complexity of the virtualized environment, which contributes to an increased risk of failure, is the fundamental factor for this worry.

Failure may be analyzed in various ways, but failure prediction is not one of them. Failure analysis methods are essential for locating the origin of hardware and software malfunctions in a cloud computing environment that is already in operation. Nevertheless, from the other perspective, the primary objective of failure detection is to identify activities that are likely to fail before they do so. In order to determine which metrics and qualities of cloud applications are the most important, a failures assessment and forecasting model is designed and put into operation. Evaluation and failure prediction both have the same overarching objective: to examine the often-changing dynamic characteristics of cloud applications to maximize the efficiency of such applications and reduce the number of unsuccessful activities.

The field of research known as machine intelligence investigates the development and study of techniques that can learn from data. These algorithms perform their tasks by constructing a model according to the inputs and then utilizing that model to generate predictions or choices instead of just carrying out the explicitly coded instructions. To guarantee that the suggested model can provide a high level of accuracy during the phase of projection, the accuracy of the classifier has been examined based on several criteria. As a result, the following is the most significant contribution that this study has made: This research and analysis on several VM logs to identify the elements of VM trace files that are related to jobs and tasks, both straggler or not.

- To design and construct a faulty node forecasting model related to using different machine learning techniques and many data nodes or VM logs.
- To develop a framework for defect prediction that may identify failure before something suddenly arises.
- Analyze the model we have presented to guarantee that the model is adaptable and can be used with a variety of VM logs. Because of this, we have used various ML classification methods on a wide variety of VM logs. Then, based on the assessment findings, we chose the most effective model to achieve the maximum possible overall accuracy.

The paper is described as, SECTION II demonstrates the literature review of a proposed model where the various existing system has been analysed and all the current systems are identified. In SECTION III, research methodology of the presented system and implementation details are described, while in SECTION IV the proposed algorithm is described in detail. SECTION V focuses on results and discussion of the experimental set-up, and various experimental results, and finally, the conclusion and recommendations are discussed in SECTION VI of the proposed model.

2. LITERATURE

The objective of the fourth revolution is to create a manufacturing environment that is real-time, sophisticated, extensible, and independent. This environment will be provided by an industrial setting. In order to make this vision a reality, recent breakthroughs in digitalization, such as the Internet of Things (IoT) and cloud computing, which have piqued the interest of researchers in both research and business and have a wide range of potential applications, will need to be implemented. In this section we describe some techniques used for detection of straggler node detection in distributed and network environments using machine learning techniques.

The outlines a methodology for the identification of straggler nodes in distributed environments by employing a machine learning approach that is based on large-scale task executed log data [1]. After the machine learning algorithm has been run through its paces, the system will forecast the list of straggler nodes and dynamically remove them from the execution list. In the beginning, a reinforcement learning algorithm that was based on Q-Learning was presented for execution and validation of the system result in multi-node contexts. a recently developed protocol called Machine Learning based secure RPL routing (MLRP) [2]. At the beginning, the MLRP protocol uses the Cooja simulator to generate a complicated dataset that includes both normal and attack behaviour. Second, the machine learns how to effectively identify attack behaviours such as version, rank, and denial of service by analysing the dataset. These attack behaviours include: (DoS). Additionally, the Support Vector Machine (SVM) classifier is used by the MRLP in order to categorise the various kinds of attacks. The dimensionality of the dataset is successfully reduced using enhanced principal component analysis (PCA), which is developed in order to increase the performance of support vector machines (SVM). In conclusion, the results of the simulation show that the proposed MLRP protocol enhances the attack detection accuracy while maintaining precision and is suitable for the context of the internet of things (IoT).

The increase in the size of the network and our reliance on it for all aspects of our lives have consequently resulted in the generation of many attacks on the network by malicious parties [3]. These attacks can be either novel or the mutations of attacks that have been used previously. These assaults provide a number of issues for network security experts, who are tasked with defending computer and network nodes, as well as the data associated with those nodes, from potential incursions. By continuously monitoring the network traffic and providing protection for the entrance points of a network, a network intrusion detection system, also known as a NIDS, is one of the most effective security solutions that can be implemented. In spite of the significant efforts put in by researchers, the National Intrusion Detection System (NIDS) continues to have a high false alarm rate (FAR) when it comes to identifying fresh threats. In this study, we offer a new NIDS framework that is based on a deep convolution neural network. This framework makes use of network spectrogram pictures that are created using the short-time Fourier transform. The CIC-IDS2017 dataset served as the basis for our evaluation of the effectiveness of the approach that we have presented. a distributed sensor-fault detection and diagnosis system based on machine learning techniques has been developed [4]. This system is designed so that the fault detection block may be embedded in the sensor so that output can be achieved immediately after data collection. This block is made up of an auto-encoder, which is responsible for converting the input signal into a lower-dimensional feature vector. This feature vector is then passed on to a Support Vector Machine (SVM), which determines if the signal is normal or incorrect. In order to lessen the computational burden placed on the sensor once an issue has been identified, the fault diagnosis is carried out at a centralised node, such as a network server. For the purpose of diagnosis in this study, a Fuzzy Deep Neural Network (FDNN) is used. This enables the provision of additional information, such as the nature of the defect. In this scenario, the input is fed into a deep neural network, and then it goes through a fuzzy representation procedure. The output of these two components is then combined using layers that have a dense network of connections. This multi-modal approach discovers high-level representations in the data that traditional approaches do not take into account. We use data acquired from a temperature-to-voltage converter that is healthy. These data are then injected with five different kinds of faults in order to evaluate how well the proposed model performs: drift faults, bias faults, accuracy deterioration faults, spike faults, and stuck faults.

According to an evaluation of the current state-of-the-art of distributed filtering and control of industrial CPSs defined by differential dynamics models [5], the phrase "state of the art" is appropriate. Sensor networks, manipulators, and power systems all get a focus of attention that is beyond the norm. In this article, three common Kalman-based distributed algorithms for real-time monitoring are described, and their performances on computation load, communication burden, and scalability are reviewed in detail. Following this, further information on the features of non-Kalman instances is provided in light of created filter structures. In addition, the most recent advancements in distributed cooperative control of mobile manipulators and distributed model predictive control in industrial automation systems are analysed and discussed. The needs of power sharing, voltage regulation, and frequency regulation are met by power systems that must use representative distributed control techniques that are characterised by controller architectures. These strategies are presented in a systematic manner using droop characteristics. A Naive Bayes classifier and a convolution neural network (CNN) to increase the convergence performance and discover the node flaws, as stated in [6]. In the end, we investigate the convex hull, Naive Bayes, and CNN algorithms by applying them to real-world datasets in order to locate and catalogue the errors. Both simulation and experimental results confirm the technique's practicality and efficiency, and demonstrate that, based on performance measures, the CNN approach contains defects that are more easily discovered than those in the convex hull algorithm. The use of wireless sensor networks to serve a broad variety of monitoring and control applications,

such as environmental surveillance, industrial sensing, or traffic checking, has become more common in recent years. WSNs are comprised of an extremely large number of tiny, low-power, wireless devices. These devices may, for instance, be used to monitor the environment, industrial sensors, or traffic. WSNs are comprised of a vast number of wireless devices that are often deployed at dispersed and haphazardly organised locations. These devices are typically tiny and have a low power output. A wide variety of mobile and unavoidable apps are constantly receiving and processing data from the real environment and providing data about the identified situation or circumstances at an extremely granular level.

An intelligent fault detection, energy-efficient, quality-of-service routing technique based on reinforcement learning was developed to find the optimal route with the least amount of end-to-end latency [7]. Other benefits of this technique include quality of service and energy efficiency. However, the selection of the cluster head is contingent on the residual energy that is produced by the cluster nodes, which in turn reduces the existence of the whole network. As a consequence of this, it lengthens the lifespan of the network, reduces the amount of energy that is used during data transmission, and increases the resiliency of the network. The findings of the experiments reveal that the effectiveness of the network has been effectively improved by fault-tolerance solutions that incorporate highly trusted computing capabilities, which has resulted in a decreased chance of network failure. A review on the MG fault detection methodologies and their limitations is presented, along with the proposal of a new discrete-wavelet transform (DWT) based probabilistic generative model to investigate the exact solution for fault diagnosis of MG [8]. The suggested model is composed of numerous layers, each of which contains a restricted Boltzmann machine (RBM). This gives the model the capacity to do probability reconstruction based on the data it receives. The individual RBM layers are trained using an unsupervised learning method, in which an artificial neural network (ANN) algorithm is used to tweak the model in order to achieve the goal of decreasing the amount of error that exists between the actual class and the one that was predicted. By altering both the input signal and the sample frequencies, the efficiency of the model that has been provided may be evaluated. The resilience of the investigated model is put to the test by introducing a certain amount of assumed noise together with the sample data.

A total of six different classifiers were used and put through their paces using WSN [9]. The results of the simulation are shown in Table 2. It is shown that the RF Classifier can recognize defects such as Offset fault, Gain fault, Out-of-Bounds fault, and Spike fault with an accuracy of 97 percent each; however, it can detect the Data loss fault with an accuracy of 99 percent. Analyses have been performed on the six different classifiers, namely the Support Vector Machine, Convolutional Neural Network, Multilayer Perceptron, Stochastic Gradient Descent, Random Forest, and Probabilistic Neural Network. The information that is produced by the sensor nodes has a variety of errors introduced into it, including an Offset fault, a Gain fault, a Stuck-at fault, an Out of Bounds fault, a Spike fault, and a Data loss fault. Classifiers do quality assurance checks on the inaccurate data. The results of the simulation demonstrate that the Random Forest discovered more errors than any other classifier in that category, and it also performed better than any of those other classifiers. In accordance with the findings of [10], an autocorrelation-assisted feature extraction approach was developed for the identification of rolling bearing failures. To achieve this goal, accelerometers were used to capture the vibration signals of healthy bearings as well as various problematic bearings. Next, autocorrelation of the individual vibration signals was performed to investigate the degree to which they were comparable to one another on a temporal scale. After this, many statistical, Hjorth, and non-linear features were extracted from the respective vibration correlograms. These features were then submitted to feature reduction using a method known as recursive feature elimination, and the results were analysed. After having their dimensions lowered, the highest-ranked feature vectors were then input into a random forest classifier so that the signals could be categorised appropriately. Numerous tests were conducted with I three distinct fault widths, (ii) four distinct shaft speeds, and (iii) two distinct sample frequencies.

In the scheme described in [11], the simulation results for feature extractions are evaluated on a standard International Electrotechnical Commission (IEC) medium voltage microgrid that is compatible with the MATLAB/SIMULINK software environment. On the other hand, Python is utilized for the training and testing of the data mining model. The outcomes are examined in grid-connected and islanded modes, for both looping and radial layouts, using a variety of fault and no-fault scenarios that were simulated. When compared to decision trees, support vector machines, and random forests, the findings demonstrate that the accuracy of the suggested logistic regression and AdaBoost classifier is much higher. a framework has been developed that uses supervised machine learning and statistical approaches in order to identify and anticipate many issues at the infrastructure-level of edge clouds [12]. The structure that has been suggested is made up of three primary components that are in charge of the following tasks: (1) data preprocessing; (2) fault detection; and (3) fault prediction. According to the findings, the framework is capable of identifying and predicting several errors online in a timely manner. For instance, by using SVM, NB, RF, and ANN models, the framework is able to identify non-fatal CPU and HDD overload problems with an F1 score that is more than 95 percent. This is possible because to the framework's ability to detect faults in real time. a machine learning-based intelligent configuration synthesis method that optimizes bandwidth usage by reproducing frames only when a connection has a greater tendency for failure is proposed. This mechanism was developed in order to improve use of available bandwidth [13]. In order to ensure the delivery of control data within a predetermined window of time, safety-critical systems used in industrial automation need real-time assurances from the Ethernet layer. As a result of this, the network design for such systems has generally been conservative. This means that the needs for the system are designed with painstaking attention to detail, and the worst-case scenario for network traffic is included into the design before it is deployed.

Wang and Huang [14] PFT is a new scheduling technique for Hadoop YARN that uses multiple level queues, time factors, task urgency factors, and domain capacity ratio to efficiently tradeoff performance and fairness while reducing the makespan of Map - Reduce tasks. We use Hadoop YARN to develop PFT as a modular scheduling. PFT can minimize the makespan of Map - Reduce jobs by 34.53 percent, enhance CPU usage by 35.93 percent, and enhance memory usage by 38.98 percent, according to empirical observations. Vorapongkitipun and Nupairoj [15] introduce New Hadoop

Archive, a methodology built on Hadoop Archive (HAR), to increase metadata storage use and boost the effectiveness of reading documents in HDFS. Furthermore, we extend the HAR abilities to give extra files to be placed into current archive files. The findings of the testing reveal that our strategy can significantly enhance the access efficiency of small documents, outperforming HAR by 85.47 percent.

Shah and Padole [16] suggest a custom block placement strategy that takes advantage of the CPU's processing power throughout block placement. This method will assist MapReduce decrease the amount of data transferred between nodes and racks. Researchers have shown that it could only send data blocks from specified files to specified points. Because the majority of the files are unaffected, this strategy has no effect on the cluster's total task scheduling. The results of the experiments show that this technique works for both homogenous and heterogeneous clusters. Kim et al. [17] examine the methods for finding the best Hadoop HDFS configuration parameters. The automatic benchmark configuration method (ABCM) has been presented and validated as an optimizing tool that uses a 2-sampling technique to minimize the cost function and cost of finding the best system parameters.

For greater performance, Mohammed and Bharati [18] recommends a dynamic slot allocation method with an optimized scheduler. Dick et al. [19] tackle the practical issues and inconsistencies encountered when operating Hadoop on LINUX CENT OS 6.6. In the Hadoop design, up to 6 data units can be deployed. The results of different RAM amounts were unanticipated, and the MapReduce process revealed various abnormalities that were not covered in the manual. Bante and Rajeswari [20] proposes a method for migrating data from relational to NoSQL (MongoDB) databases and doing bigdata analysis with Hadoop Map - Reduce on top of HDFS. Sqoop can also be used for migration of info from relational database systems to Hadoop for analytics. Hive is a tool that allows you to move data obtained from Hadoop to MongoDB. Mallika and Selvamuthukumaran [21] described the upgraded cloud with Hadoop structure's capability and preparation time. Zhou and Wen [22] presented the technology can more correctly determine the maximum load for every DataNode for Hadoop, and it can instantly switch the maximum load of Hadoop distributed network while the user asks a document again. Jena et al. [23] seeks to alleviate the strain on the datanode in the HADOOP design by offering help via the aggregator node, that interoperates amongst the name and the data node. Kalimoldayev et al. [24] provides the findings of experimental tests carried out to achieve the main goal of designing and implementing a revolutionary cyclic MapReduce structure based on Hadoop architecture.

Apichanukul et al. [25] using coding, clustering, as well as adaptive selection, researchers investigated the trade-off between wallclock duration, networking, and processing needs for gradient-based dispersed training. Bangare et al. [26-29], Shelke et al. [30], Gupta et al. [31], Awate et al. [32] and Pande et al. [33-35] have worked in the area of the machine learning and IOT issues etc. Stragglers benefit both from coding as well as clustering, whereas adaptive selection aims to minimize computational and communication demands. Two unique methods are suggested that try to combine the advantages of both methods. The benefit of straggler avoidance by coding doesn't really pay for the additional compute load, even within terms of wall-clock duration, whenever the dispersion of the labors' computation durations has a small tail, according to data and quantitative data. For overcoming job stragglers and performance deterioration in cluster system for huge data processing, current method includes speculative & replica implementation. Basic concept of straggler and ML are referred from [36, 37].

The above literature describes work done by previous researchers but still few systems having some gaps such as high error rate, module overfitting problem when it deals with supervised classifiers and insufficient knowledge extraction from log data.

3. PROPOSED SYSTEM DESIGN

Figure 1 demonstrates a proposed system for classifying and predicting faulty in a highly HDFS environment. In a conventional cluster system where the server stack contains multiple servers installed on virtual machines or individual boxes, the classification and the faults detection are mainly achieved using the static heartbeat detection mechanism. In heartbeat detection mechanism a process periodically sends a request to all the other servers which indicates liveness of the servers. For the proposed system the data was evaluated using the rules & norms established while preprocessing. Every attribute has a minimum and maximum constraint for certain variables, and the program kills the object if one of the other values surpasses or exceeds the limits. The threshold values of these attributes are decided based on the node configurations like how many CPU core it has, how much memory is allocated to the node and what is the network bandwidth on that node. Pre-processing includes acquiring data, cleansing it, filtering it, and standardizing it. Cleaning and mending false or inaccurate data from files, records, or sets comprises discovering and restoring lost, erroneous, or illogical information, as well as restoring, upgrading, or deleting filthy or critical information. To cleanse information dynamically, we referred web services of an open-source scripting tools OpenRefine. As it is an open-source tool, we have customized the code according to our needs. This tool also transforms the data from one format to another. It makes sure that the data is cleanly structured. To normalize the data, we used comparable sampling processes and screened the standard dataset to remove the wrongly classified events. As a part of sampling process, we initially identified the target data that needs to process which are log files on the virtual machine boxes. These log file contains some information which is irrelevant to our research. To filter out such information sampling frame is created by identifying the parameters like CPU, memory, and network usage from the logs. These parameters are then compared with the predefined data set to filter out the information. The ordinary and numeric values from textual information are retrieved during extracting the features. TF-IDF, relational features, feature dependencies as well as Co-relation co-occurrence must all be extracted from the dataset obtained using multiple techniques. TF-IDF term frequency-inverse document frequency technique is used to achieve the text classification from the log files. The words like GC, Avg, Sum, Heap, worker that occur more frequently in one log file and less frequently in other log file has given more importance as they are more useful for classification. LDA (Linear Discriminant Analysis) dimensionality reduction technique is used to find most important feature from the log files. Ultimately, using a supervised classification method, the method recognizes every record as assault or normal. In

addition, the system shows an unidentified attack categorization of a standard real - world dataset.

We used SVM as just a supervised classification technique for identification and tracking. After that, the classifiers are trained using machine learning approach. Information with class labels is presented at the start. Various decision trees are generated utilizing randomized characteristics from the set of features as well as the majority outcome class of all of the decision trees is chosen as the result of the SVM method to assess identity deception on social media. The F1-Score as well as precision, recall and accuracy were calculated after the findings were analyzed using the confusion matrix.

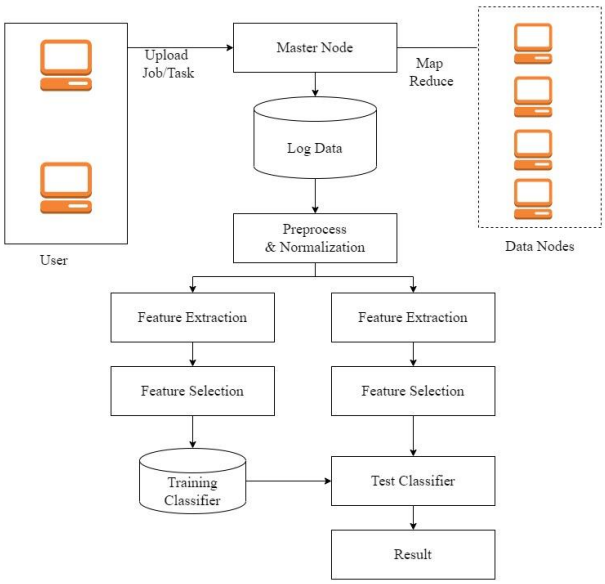

Figure 1. Proposed system architecture

4. ALGORITHM DESIGNING

The above algorithm technique presents a solution that gets around the problem of straggler nodes being predicted, and it does so use the supervised classification method. The total values for the retrieved parameters are shown by the proposed hybrid technique. These values are calculated using a variety of machine learning algorithms. The following methodology was used in order to construct the machine performance report in accordance with the combination reinforcement learning based prediction method.

Input: Normalized training dataset Train_Data[], Normalized testing dataset Test_Data[], defined threshold qTh

Output: Result set as output with {Predicted_class, weight}

(1) Read all test data from *Test_Data*[] using below function for validating to training rules, the data is normalized and transformed according to algorithms requirements

$$\text{test\_Feature(data)} = \sum_{m=1}^{n} (.\ \text{Attribute\_Set[A[m] ... . A[n]} \leftarrow Test\_Data)$$

(2) Select the features from extracted attributes set of test_Feature(data) and generate feature map using below function.

$$\text{Test\_FeatureMap [t ... . n]} = \sum_{x=1}^{n} (t) \leftarrow \text{test\_Feature(x)}$$

Test_FeatureMap [x] are the selected features in pooling layer. The convolutional layer which is the main component of CNN where most of the computation happens. Features are extracted from test log data as input and filters using the CNN. It then identifies wide range of patterns in the test log files. It generates an output value that indicates how similar the input region is to the filtered pattern. ReLU is used as activation function. Output of the convolution layer is passed to pooling layer and those selected features are stored in *Test_FeatureMap.*

(3) Now read entire taring dataset to build the hidden layer for classification of entire test data in sense layer

$$\text{train\_Feature(data)} = \sum_{m=1}^{n} (.\ \text{Attribute\_Set[A[m] ... . A[n]} \leftarrow Train\_Data)$$

(4) Generate the training map using below function from input dataset

$$\text{Train\_FeatureMap [t ... . n]} = \sum_{x=1}^{n} (t) \leftarrow \text{train\_Feature}(x)$$

Train_FeatureMap[t] is the hidden layer map that generates feature vector for build the hidden layer that evaluate the entire test instances with train data. Rectified Linear Activation (ReLU) is used as an activation function in the hidden layer. It is simple to implement and effective as compared to other activation functions. The ReLU returns the value provided as input directly, or the value 0 if the input is 0 or less. We can state this function *f()* mathematically using the *max()* function over the set of 0 and the input i as below.

$$f(i) = max\{0, i\}$$

(5) After generating the feature map we calculate similarity weight for all instances in dense layer between selected features in pooling layer.

$$\text{Gen\_weight} = \text{CalcWeight(Test\_FeatureMap} \,||\, \sum_{i=1}^{n} \text{Train\_FeatureMap[i])}$$

(6) Evaluate the current weight with desired threshold

$$if(Gen\_weight >= qTh)$$

(7) Out_List.add**(**trainF.class, weight**)**

(8) Go to step 1 and continue when Test_Data==null

(9) Return Out_List

This composite classification method receives input in the form of all potential outcomes produced by specified data mining algorithms. These results are then fed as input to the composite classification algorithm. As per the given values of probabilities method in step 5, it creates the exact weight for a particular virtual environment, and on the basis of that, and we may estimate the potential of straggler for VM.

5. RESULTS AND DISCUSSIONS

The systems were deployed on the widow's platform with JDK 1.7 and the Weka 3.7 machine learning framework for an

extended experimental examination. In this experiment, we used a real-time node log file dataset to illustrate the classification accuracy of SVM. Robust repeated k-Fold cross-validation technique is used to cross validate the dataset. In this technique the log model is trained k times where k is the fold. First the log data set is split into 2 sets and the process is repeated for 20 folds. For every repetition the model data is trained and then used to validate the log data set. For every repetition the result is saved and then the average is calculated to get the result. Figure 2 describes the results of similar experiments utilizing various cross validation techniques. According to the findings, 15-fold cross validation has the highest average classification accuracy of 96.70% with SVM and its activation function, the 5-fold cross validation likewise obtains 92.50%. Figure 2 shows the results of a 10-fold data cross validation. It has comparable level of accuracy as that of 5-fold cross validation.

The Figure 2 describes a performance evaluation of proposed SVM classification with various cross fold validation such as 5-fold,10-fold and 15-fold respectively. The 15-fold provides highest precision as 93.10% while 10-fold obtains lower precision 92.0%.

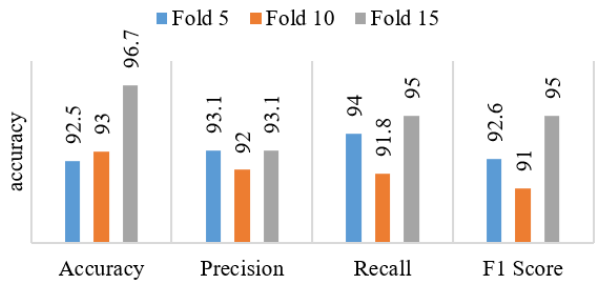

Figure 2. System validation with various cross validation using proposed framework for faulty node detection

Figure 3 describes a straggler node detection and prediction accuracy using proposed hybrid machine learning algorithms. The five different machine learning (ML) techniques has used for such as NB, RF, ANN, ADABOOST and SVM. The SVM produces highest accuracy The NB gives 85.1% accuracy; RF produces 91.3%, ADABOOST produces 88.1% accuracy, ANN produces 87.4% whereas SVM gives 96.7% of highest accuracy amongst all.

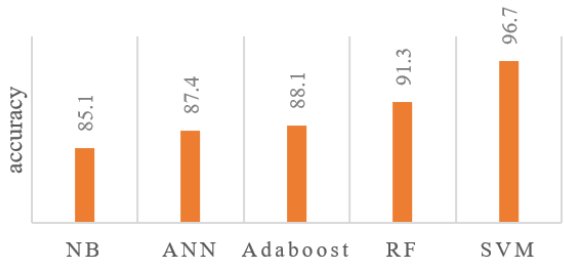

Figure 3. Comparative analysis for faulty node detection with various machine learning algorithms

6. CONCLUSIONS

This paper proposed the detection of the faulty node in the Hadoop distributed file system using supervised classification techniques. The four data nodes and a single master node have been deployed to evaluate the system. All virtual machines are used as input nodes for model building. This lock data describes the current node health, whether it is in idle mode aur heated. The entire log data should contain some missing values that are eliminated using numerous data normalization and filtration techniques. Various feature extraction and selection methods are applied to extract the unique features and robust model building. Different supervised classification algorithms are applied to real data sets such as random forest, Naive Bayes, artificial neural network, AdaBoost and support vector machine. As a result, the SVM produces higher accuracy over the other conventional machine learning classification algorithms. Collect a data from the large distributed environment in an unstructured format and applying deep learning algorithms will be an interesting future work for this research. In addition to the aforementioned contributions, future research will include integration of the suggested deep learning techniques into cluster work schedule decision-making elements such as the task scheduler in distributed systems, as well as improving its ability to handle limitation situations when no tasks are forwarded.

REFERENCES

[1] Gaykar, R.S., Nalini, C., Joshi, S.D. (2021). Identification of straggler node in distributed environment using soft computing algorithms. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), pp. 1-5. https://doi.org/10.1109/ESCI50559.2021.9396825

[2] Momand, M.D., Mohsin, M.K. (2021). Machine learning-based multiple attack detection in RPL over IoT. In 2021 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-8. https://doi.org/10.1109/ICCCI50826.2021.9402388

[3] Khan, A.S., Ahmad, Z., Abdullah, J., Ahmad, F. (2021). A spectrogram image-based network anomaly detection system using deep convolutional neural network. IEEE Access, 9: 87079-87093. https://doi.org/10.1109/ACCESS.2021.3088149

[4] Jan, S.U., Lee, Y.D., Koo, I.S. (2021). A distributed sensor-fault detection and diagnosis framework using machine learning. Information Sciences, 547: 777-796. https://doi.org/10.1016/j.ins.2020.08.068

[5] Ding, D., Han, Q.L., Wang, Z., Ge, X. (2019). A survey on model-based distributed control and filtering for industrial cyber-physical systems. IEEE Transactions on Industrial Informatics, 15(5): 2483-2499. https://doi.org/10.1109/TII.2019.2905295

[6] Regin, R., Rajest, S.S., Singh, B. (2021). Fault detection in wireless sensor network based on deep learning algorithms. EAI Endorsed Transactions on Scalable Information Systems, 8(32): e8-e8. https://doi.org/10.4108/eai.3-5-2021.169578

[7] Mahmood, T., Li, J., Pei, Y., Akhtar, F., Butt, S.A., Ditta, A., Qureshi, S. (2022). An intelligent fault detection approach based on reinforcement learning system in wireless sensor network. The Journal of Supercomputing, 78(3): 3646-3675. https://doi.org/10.1007/s11227-021-04001-1

[8] Rahman Fahim, S., Sarker, S., Muyeen, S.M., Sheikh, M., Islam, R., Das, S.K. (2020). Microgrid fault detection

and classification: Machine learning based approach, comparison, and reviews. Energies, 13(13): 3460.
[9] Gnanavel, S., Sreekrishna, M., Mani, V., et al. (2022). Analysis of fault classifiers to detect the faults and node failures in a wireless sensor network. Electronics, 11(10): 1609. https://doi.org/10.3390/electronics11101609
[10] Roy, S.S., Dey, S., Chatterjee, S. (2020). Autocorrelation aided random forest classifier-based bearing fault detection framework. IEEE Sensors Journal, 20(18): 10792-10800. https://doi.org/10.1109/JSEN.2020.2995109
[11] Baloch, S., Muhammad, M.S. (2021). An intelligent data mining-based fault detection and classification strategy for microgrid. IEEE Access, 9: 22470-22479. https://doi.org/10.1109/ACCESS.2021.3056534
[12] Soualhia, M., Fu, C., Khomh, F. (2019). Infrastructure fault detection and prediction in edge cloud environments. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, pp. 222-235. https://doi.org/10.1145/3318216.3363305
[13] Desai, N., Punnekkat, S. (2020). Enhancing fault detection in time sensitive networks using machine learning. In 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, pp. 714-719. https://doi.org/10.1109/COMSNETS48256.2020.9027357
[14] Wang, Q., Huang, X. (2016). PFT: A performance-fairness scheduler on Hadoop yarn. In 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 76-80. https://doi.org/10.1109/ICSESS.2016.7883019
[15] Vorapongkitipun, C., Nupairoj, N. (2014). Improving performance of small-file accessing in Hadoop. In 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 200-205. https://doi.org/10.1109/JCSSE.2014.6841867
[16] Shah, A., Padole, M. (2018). Load balancing through block rearrangement policy for Hadoop heterogeneous cluster. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 230-236. https://doi.org/10.1109/ICACCI.2018.8554404
[17] Kim, J., Park, N. (2015). Identification of the optimal Hadoop configuration parameters set for MapReduce computing. In 2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence, pp. 108-11. https://doi.org/10.1109/ACIT-CSI.2015.27
[18] Mohammed, A.Q., Bharati, R. (2017). An efficient technique to improve resources utilization for Hadoop MapReduce in heterogeneous system. In 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 12-16. https://doi.org/10.1109/INTELCCT.2017.8324012
[19] Dick, M., Ji, J.G., Kwon, Y. (2017). Practical difficulties and anomalies in running Hadoop. In 2017 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 1501-1504. https://doi.org/10.1109/CSCI.2017.263
[20] Bante, P.M., Rajeswari, K. (2017). Big data analytics using Hadoop map reduce framework and data migration process. In 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, pp. 1-5. https://doi.org/10.1109/ICCUBEA.2017.8463824
[21] Mallika, C., Selvamuthukumaran, S. (2017). Hadoop framework: Analyzes workload predicition of data from cloud computing. In 2017 International Conference on IoT and Application (ICIOT), pp. 1-6. https://doi.org/10.1109/ICIOTA.2017.8073624
[22] Zhou, H., Wen, Q. (2014). Load balancing solution based on AHP for Hadoop. In 2014 IEEE Workshop on Electronics, Computer and Applications, pp. 633-636. https://doi.org/10.1109/IWECA.2014.6845699
[23] Jena, B., Gourisaria, M.K., Rautaray, S.S., Pandey, M. (2017). Name node performance enlarging by aggregator based HADOOP framework. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 112-116. https://doi.org/10.1109/I-SMAC.2017.8058320
[24] Kalimoldayev, M.N., Siladi, V., Satymbekov, M.N., Naizabayeva, L. (2017). Solving mean-shift clustering using MapReduce Hadoop. In 2017 IEEE 14th International Scientific Conference on Informatics, pp. 164-167. https://doi.org/10.1109/INFORMATICS.2017.8327240
[25] Apichanukul, W., Kawahara, J., Kasahara, S. (2016). Accuracy improvement for backup tasks in Hadoop speculative algorithm. In 2016 IEEE International Conference on Computer and Information Technology (CIT), pp. 500-507. https://doi.org/10.1109/CIT.2016.17
[26] Bangare, S.L. (2022). Classification of optimal brain tissue using dynamic region growing and fuzzy min-max neural network in brain magnetic resonance images. Neuroscience Informatics, 2(3): 100019. https://doi.org/10.1016/j.neuri.2021.100019
[27] Bangare, S.L., Dubal, A., Bangare, P.S., Patil, S.T. (2015). Reviewing Otsu's method for image thresholding. International Journal of Applied Engineering Research, 10(9): 21777-21783. https://dx.doi.org/10.37622/IJAER/10.9.2015.21777-21783
[28] Bangare, S.L., Pradeepini, G., Patil, S.T. (2018). Regenerative pixel mode and tumor locus algorithm development for brain tumor analysis: A new computational technique for precise medical imaging. International Journal of Biomedical Engineering and Technology, Inderscience, 27(1/2). https://www.inderscienceonline.com/doi/pdf/10.1504/IJBET.2018.093087
[29] Bangare, S.L., Pradeepini, G., Patil, S.T. (2017). Neuroendoscopy adapter module development for better brain tumor image visualization. International Journal of Electrical & Computer Engineering, 7(6): 3643-3654. http://ijece.iaescore.com/index.php/IJECE/article/view/8733/7392.
[30] Shelke, N., Chaudhury, S., Chakrabarti, S., Bangare, S. L., Yogapriya, G., Pandey, P. (2022). An efficient way of text-based emotion analysis from social media using LRA-DNN. Neuroscience Informatics, 100048. https://doi.org/10.1016/j.neuri.2022.100048
[31] Gupta, S., Kumar, S., Bangare, S.L., Nuhmani, S., Alguno, A.C., Samori, I.A. (2022). Homogeneous decision community extraction based on end-user mental behavior on social media. Computational Intelligence

and Neuroscience, 2022: 3490860. https://doi.org/10.1155/2022/3490860
[32] Awate, G., Bangare, S., Pradeepini, G., Patil, S. (2018). Detection of Alzheimers disease from MRI using convolutional neural network with tensorflow. arXiv preprint arXiv:1806.10170. https://doi.org/10.48550/arXiv.1806.10170
[33] Pande, S.D., Chetty, M.S.R. (2018). Analysis of capsule network (Capsnet) architectures and applications. J Adv Res Dynam Control Syst, 10(10): 2765-2771.
[34] Pande, S.D., Chetty, M.S.R. (2019). Position invariant spline curve based image retrieval using control points. Int J Intell Eng Syst, 12(4): 177-191.
[35] Pande, S.D., Patil, U.A., Chinchore, R., Chetty, M.S.R. (2019). Precise approach for modified 2 stage algorithm to find control points of cubic Bezier curve. In 2019 5th International Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1-8. https://doi.org/10.1109/ICCUBEA47591.2019.9128550
[36] Gaykar, R.S., Khanaa, V., Joshi, S.D. (2021). Detection of faulty nodes in distributed environment using machine learning. In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 228-232. https://doi.org/10.1109/ICAC3N53548.2021.9725478
[37] Gaykar, R.S., Khanaa, V., Joshi, S.D. (2022). A hybrid supervised learning approach for detection and mitigation of job failure with virtual machines in distributed environments. Ingénierie des Systèmes d'Information, 27(4): 621-627. https://doi.org/10.18280/isi.270412