Cluster Computing
https://doi.org/10.1007/s10586-018-2015-4

WFCM based big sensor data error detection and correction in wireless sensor network

R. Sheeba[1] · G. Jiji[1] · T. Ajith Bosco Raj[1]

Received: 20 December 2017 / Revised: 24 January 2018 / Accepted: 2 February 2018

Abstract

In WSN the requested data is collected from the initial node i.e. sender and the information are uploaded on a cloud platform. Only numeric Data type is considered in this error detection and correction technique. Map Reduce algorithm is applied on clusters made by big data and Weighted Fuzzy C-Means Clustering (WFCM) technique is used for clustering. Completely different operations are performed on the cloud platform like error detection, location finding, data cleansing and error recovery. Throughout the filtering of big data sets, whenever an abnormal knowledge is encountered, detection rule has to perform two tasks. *"fd (n/e,t)"* is decision making function. It is used to determine whether the detected anomalous data is a true error. In other words, *fd (n/e,t)* has two outputs, "false negative" for detecting a true error and "false positive" to select non-error data. *"fl (n/e,t)"* is a function for tracking and returning original error source.

Keywords Map reduce · Weighted fuzzy C-means clustering · Error detection · Error localization · Kernel SVM

1 Introduction

The wireless sensor networks (WSN) has several independent wireless sensor nodes connected with each other to form a network. Each and every individual sensor node is capable of sensing and processing information [1]. The WSN monitors and interacts with people's physical environment [2] and the collected data is hopped to the requested node through the gateway. With the tremendous improvement in technology where daily life starts with sharing of data, the data collected from the Sensor nodes collectively form the Big Data. Big data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal.

WSN consist of a large numbers of wireless sensor nodes dispersed in one or more base stations, where big sensor data is collected. While transforming the information in sensor network, loss of big sensor data or error may be spotted in the received data [14]. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, error detection and correction, updating and information privacy.

Big sensor data has five characteristics such as volume, variety, veracity, velocity and value which are known as 5 V's of big data. The big data is collected from several areas such as meteorology, connectomics, complex physics simulations, genomics, biological study, gene analysis and environmental research [4]. These collections can start from complex framework structures such as boundless scale sensor frameworks and relational association [9].

The big data collected from the Wireless Sensor Network is called as the big sensor data. The big sensor data can be easily corrupted and lost, due to the presence of hardware faults and inaccuracies in nodes which may occur naturally or by intrusion [6, 12]. In real time network application, the collected big data can be abnormal and errors might occur [15]. Specifically numerical data errors are set down and introduced in big data [7]

But the big sensor data collected from WSN has to be clean, accurate, error free and of less loss for an efficient decision making [3]. Therefore, Big sensor data error has to be detected and corrected in an efficient way which is a challenging one [5]. For the error detection process, initially the errors are identified by error classification

✉ R. Sheeba
rsheebaphd@gmail.com

[1] Department of ECE, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu 627 152, India

Published online: 16 February 2018

method. Classification can be done using several algorithms for numerical errors [13].

In Big Sensor Data, error detection often requires real time processing and storage for massive sensor data which uses the complex error model to detect the event of abnormality [10]. It deals with context of using inherently complex error models to spot and locate events of abnormalities [11].

The map reduce error detection approach is usually used in big sensor data for finding errors in data sets [8]. The defined error model will trigger the error detection process, compares the result with the previous error detection methods of sensor network systems.

2 Litrature survey

José Carlos et al. [16] examined fault detection and diagnosis (FDD) based on weightless neural networks (WNN) with applications in univariate and multivariate dynamic systems. The proposed system executed the selection of attributes in the multivariable case and did the time series mapping of the data. In the intermediate stage, the WNN performed the detection and diagnosis per class. The network outputs were then passed through a clustering filter in the final stage of the system, if a diagnosis per fault groups was necessary. The system was tested with two case studies: one was an actual application for the temperature monitoring of a sales gas compressor in a natural gas processing unit; and the other one used the simulated data for an industrial plant. The results showed the efficiency of the proposed systems for FDD with classification accuracies of up to 98.78 and 99.47% for the respective applications.

Cai et al. [17] described that Bayesian Network (BN) was a commonly used tool in probabilistic reasoning of uncertainty in industrial processes. Motivated by reduction of the overall complexities of BNs for fault diagnosis, and the reporting of faults that immediately occur, a real-time fault diagnosis methodology of complex systems with repetitive structures is proposed using Object-Oriented Bayesian networks (OOBNs). The modeling methodology consists of two main phases: an off-line OOBN construction phase and an on-line fault diagnosis phase. In the off-line phase, sensor historical data and expert knowledge are collected and processed to determine the faults and symptoms, and OOBN-based fault diagnosis models were developed subsequently. In the on-line phase, operator experience and sensor real-time data were placed in the OOBNs to perform the fault diagnosis. According to engineering experience, the judgment rules were defined to obtain the fault diagnosis results.

Chun et al. [18] recommended the wireless sensors operating in harsh environments had the potential to be error-prone. A distributive model-based diagnosis algorithm that identifies nonlinear sensor faults had been presented. The diagnosis algorithm has advantages over existing fault diagnosis methods such as centralized model-based and distributive model-free methods. An algorithm was presented for detecting common non-linearity faults without using reference sensors. The study introduced a model-based fault diagnosis framework that was implemented within a pair of wireless sensors. The detection of sensor nonlinearities was shown to be equivalent to solve the Largest Empty Rectangle (LER) problem, given a set of features extracted from an analysis of sensor outputs. A low-complexity algorithm that gave an approximate solution to the LER problem was proposed for embedment in resource constrained wireless sensors. By solving the LER problem, sensors corrupted by nonlinearity faults can be isolated and identified. Extensive analysis evaluated the performance of the proposed algorithm through simulation.

Feng et al. [19] described with the fast development of electronics and wireless communication technologies in recent years. Intelligent wireless sensor nodes were becoming increasingly popular in the online machinery condition monitoring systems. From this bring a number of benefits, such as reduced investment on the installation and maintenance of expensive communication cables, ease of deployment and upgrading. For the condition monitoring of dynamic signals, distributed computation on wireless sensor nodes is getting popular. Wireless sensor nodes are becoming more computation powerful and power efficient. As a widely recognized algorithm for bearing fault diagnosis, envelope analysis had been previously proved suitable for being embedded on the wireless sensor nodes to effectively extract fault features from common machinery components such as bearings and gears. As a continuation several envelope detection methods, including Hilbert transform, spectral correlation, band-pass squared rectifier and short-time RMS were studied. Regarding to the fact that only low frequency components in the bearing envelope is of interest, spectral correlation could be simplified for fast calculation and short-time RMS method could be considered as a simplified band-pass squared rectifier, in which partial aliasing was allowed. Thereafter, spectral correlation and short-time RMS are employed to speed up the calculation of envelope analysis on a wireless sensor node, which thereafter provided the potential to reduce power consumption of wireless sensor nodes. The computation speed comparison showed that the spectral correlation method and short-time RMS can speed up the computation speed by more than two times and five times in comparison with the Hilbert transform method. The simulation study showed that spectral correlation and short-time RMS based methods achieves similar level of accuracy as Hilbert transform. Furthermore, the experimental study showed that spectral correlation and short-time RMS based methods can well reveal the simulated three types of bearing faults while with the computation speed significantly improved.

Guesmi et al. [20] have proposed that online induction machine faults diagnosis was a concern to guarantee the overall production process efficiency. Nowadays, the industry demands the integration of smart wireless sensors networks (WSN) to improve the fault detection in order to reduce cost, maintenance and power consumption. Induction motors can develop one or more faults at the same time that could produce severe damages. The origin of most recurrent faults in rotary machines is in the components: stator, rotor, bearing and others. A novel methodology for the online faults diagnosis in induction motors is experimented. The technique uses smart WSN to obtain the machine condition based on the motor stator current analysis. The implementation of the proposed smart sensor methodology allowed the system to perform online fault detection in a fully automated way. Simulation results were presented to show the efficiency of the proposed method to detect simple and multiple faults in induction machine. It provides detailed analysis to address the challenges in designing and deploying WSNs in industrial environments, and its reliability. Suresh et al. [21] have described about security in cloud based environment.

Panda et al. [22] has described that distributed fault detection in wireless sensor network was an important problem where every sensor node identifies its own fault status based on the information from its neighboring sensor nodes. A novel distributed fault detection algorithm to detect the soft faulty sensor nodes in sparse wireless sensor networks is presented. In the proposed scheme, every sensor node gathered the information only from their neighboring nodes in order to reduce the communication overhead. The Neyman–Pearson testing method was used to predict the fault status of each sensor node and the neighboring sensor nodes. A voting scheme was applied on the fault status information to obtain the final fault status of each sensor node. The generic parameters such as detection accuracy, false alarm rate, time complexity, message complexity, detection latency, network life time and energy consumption were considered to be evaluated and the performance of the proposed scheme had been studied analytically as well as through simulation. The result showed that the proposed scheme significantly improves the performance over the existing algorithms.

Zhiyang et al. [23] recommended to consider a novel fault diagnosis mechanism for wireless sensor networks (WSNs). Without additional agents, the built-in and self-organized diagnosis mechanism can monitor each node in real time and identify faulty nodes. As the diagnosis was operated within a cluster of nodes, it could reduce the power consumption and communication traffic. A model of the diagnosis algorithm for WSNs, with a probabilistic analysis of the local and global performance is presented. Extensive experiments demonstrate the effectiveness of the proposed method.

Alic et al. [24] describes the error correction method in Next Generation Sequencing (NGS) data. Error correction strategies are separated into three classes: k range based, addition cluster/tier based and Multiple Sequence Alignment (MSA) based. A method called Muffinec is introduced as an indel aware multi-technology correction method for NGS data.

Weng et al. [25] demonstrated to accomplish both the adaptability and close universally ideal results for bad data, topology error detection and recognition issues, by leading completely distributed algorithms over convexified issue definitions. To diminish the unwinding error in convexification technique, an atomic standard punishment was added to recognize unique issues. At last, Weng et al. proposed another metric to assess discovery and recognizable proof results, which empowered a framework administrator to describe certainty for further framework operations.

3 Problem identification

Wireless sensor networks have some problems. Some of the problems are given below.

- In wireless sensor hacker can enter and access all the user information easily.
- Wireless sensor network has lower speed when compared to wired network.
- Less secure because hackers can enter the access point and obtain all the information.
- Wireless sensor networks are public frequency network and its interface is to be used for official private information.
- It has difficult setup so that signals are prone to be disrupted by the infrared and radio signals.
- Wireless networks can be accessed by any computer within range of the network’s signal and so information transmitted through the network including encrypted information may be intercepted by unauthorized users.
- Wireless sensor networks are distracted by other wireless devices.
- Nodes need to be charged at regular intervals. Battery life of the nodes is very low.
- In wireless sensor networks, much less power is consumed in processing data than transmitting it.

– A sensor network consists of a large set of sensor nodes. It follows that the cost of an individual node is critical to the overall financial metric of the sensor network.

4 Proposed methodology

The sensing elements data values are collected and processed, wherever tend to do error detection. For finding errors, spatial and temporal correlation models are approached. The error shows a sign of quick Error Recovery and Map reduce is adopted as a trendy technique. The projected error detection approaches are going to be classified based on error varieties. Specifically for numerical data errors square measure is set down and introduced in big data error detection approach.

The outlined error model can trigger the error detection method, compared to previous error detection of sensor network systems. It is designed and developed by utilizing the large processing capability. Additionally, the design feature of advanced networks also be analyzed to mix the parallel computing with an additional efficient method. Error Detection victimization Kernel SVM, is used to train the data that is received from the sensors and then it detects incorrect data. It increases its responsibility and updates the training data. Error detection is done with new data and KSVM's results are accustomed to improve the correct knowledge collection.

Figure 1 shows the flow of the proposed big sensor data error detection and correction methodology.

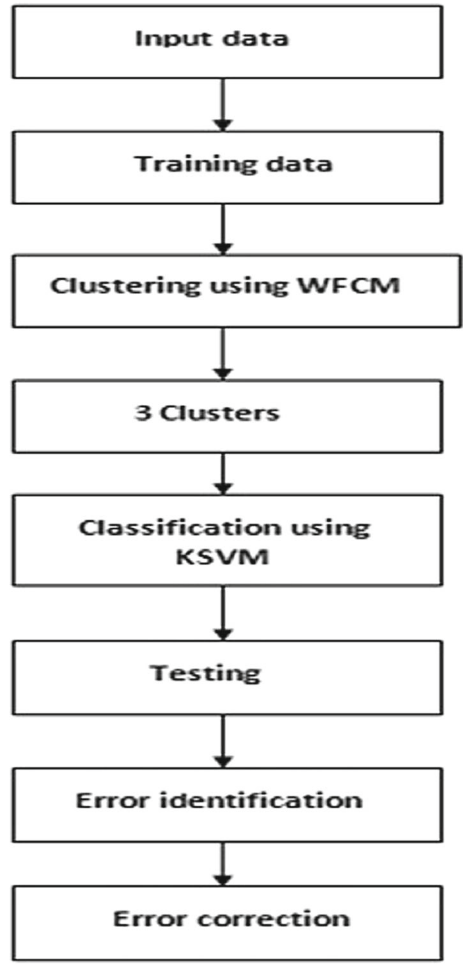

Fig. 1 Proposed big sensor data error detection and correction

4.1 Training

HDFS is a scaling portable and distributed file system written in Java for the Hadoop framework and it is the file system component of Hadoop.

It stores file system metadata and application data alone. As in the case of other distributed file systems, like Lustre, GFS and PVFS, HDFS stores metadata on dedicated servers called the Name Node. Application data is stored on other servers called the Data Node. All servers are fully communicated and connected with each other using TCP based protocols. Default HDFS stores three separate copies of the all the data block to ensure reliability, availability, and performance. In large clusters these replicas are spread across different physical racks.

So, HDFS is flexible towards two common failure scenarios: individual data node breakdown and failures that combine an entire rack offline. Replicating blocks across physical machines also increases opportunities to share, locate data processing in the time table of Map Reduce jobs, since many of the copies yield more opportunities of exploitation locality.

4.2 Mapping

Map Reduce records the isolated tasks using the Mappers. The output of the Mappers is then got together into the second set of tasks named the Reducers, where outputs from various Mappers can be joined together. Map Reduce has to be usually readily separated into independent subtasks that can be treated in parallel. The Map and Reduce function are both specialized in terms of the date structured in key and value pairs. Each reducer just treats and receives information for one specific key at a time and outputs the data it treats as a Key, Value pairs. The Hadoop Map Reduce engine has Job Tracker and one of many Task Trackers.

A Map Reduce work has to be managed by job trackers which then separate the jobs into tasks processed by the task trackers. Job Trackers sends jobs and splits to mappers or reducers as each step finishes. Trackers implement task send by the Job Tracker and reports rank to Job Trackers.

Mapping step: The master node gets the input and splits it into smaller sub problems, then dispenses it to the worker nodes that may do this again in turn which leads to a multi level structure of tree. The worker node passes the answer block to its master node. The master node then collects the answers to all the sub problems and joins them in some way to form the output of the problem. Here consider 1000 data and that some of the data's are considered as training data, then it will be clustered with the aid of WFCM.

4.2.1 Weighted fuzzy C-means clustering

The W-FCM algorithm follows an iterative optimization similar to FCM, and consequently it is affected by some of its strengths, such as its convergence in a finite number of iterations. Also, the same applies to the weights, creating the possibility that these could be far from representing the relevancy of features. By considering feature importance, WFCM algorithm is indicated as follows,

$$J(W,\ U,\ V) = \sum_{k=1}^{c}\sum_{i=1}^{n} u_{k,i}^{2}\left(d_{k,i}^{(W)}\right)^{2} \tag{1}$$

where $d_{k,i}^{(W)}$is computed by:

$$d_{k,i}^{(W)} = \sqrt{\sum_{j=1}^{d} W^{\beta}(x_{i,j} - v_{k,j})^{2}} \tag{2}$$

Minimizing Eq. (3) v_k *and* $u_{k,i}$ is indicated as follows,

$$v_k = \sum_{i=1}^{n}(u_{k,i})^{m}x_i / \sum_{i=1}^{n}(u_{k,i})^{m},\ \forall k = 1,\ldots c \tag{3}$$

$$u_{k,i} = 1 \Bigg/ \sum_{k=1}^{c}\left(d_{k,i}^{(w)} \Big/ {}_{k,\bar{k}}^{(w)}\right)^{2/m-1} \tag{4}$$

The main steps of weighted fuzzy c means clustering algorithm (W-FCM) is given as follows:Initialize the quantity of cluster as$c$, where $2 \le c \le n$ and the fuzzy separation matrix as U through an unsystematic value that it gratify the situation

$$\sum_{k=1}^{c} u_{k,j} = 1\, \forall j\, and\, 0 < \sum_{i=1}^{n} u_{k,i} < n\, \forall k \tag{5}$$

Initialize the weighting vector W with a random value such that it satisfies conditions (7) and (8).

Calculate the fuzzy centers v_k using (4).

Modernize the fuzzy partition matrix U with (5).

Update the weighting vector W

$$W_j = \begin{cases} 0 \quad if D_j = 0 \\ \dfrac{1}{\sum_{t=1}^{h}\left[\dfrac{D_j}{D_t}\right]^{\frac{1}{B-1}}} if \quad D_j \neq 0 \end{cases} \tag{6}$$

$$D_j = \sum_{k=1}^{k}\sum_{i=1}^{n} u_{k,i}(x_{i,j} - v_{k,j})^{2} \tag{7}$$

Replicate the process until the execution principle is fulfilled.

Map Reduce algorithm is applied on clusters that are made by big data here WFCM technique is used for clustering.

4.3 Kernel based support vector machine for classification

The mobile commerce system predicts a user's next purchase behavior with the aid of kernel based SVM. Then, the finest attributes are delivered to the fusion kernel support vector machine for categorization. Now, the chosen attribute from the previous progress is efficiently engaged for the isolation of two modules. For the principle of processing the non-linear procedure, the kernel functions are stated in the SVM categorization. There are two very important phases in the SVM procedure, the training phase and the effortless phase.

Training phase The output of attribute choice is provided as the input for the preparation stage. The input utility supplies the group of values which cannot be alienated. Approximately each and every one of the probable isolation of the position places are comprehended by a hectic plane. In the Lagrange pattern, it is probable to put the partition of the hectic plane standard vector during the divergent kernel task. In this association, a kernel symbolizes a few tasks, which communicates to a dot product for definite kind of attribute recording. Yet, recording a position in a better quality dimensional gap is probable to direct unnecessary assessment period and enormous storage requirements. In concrete, an original kernel task is initiated which is competent of openly estimating the dot product in the better-quality dimensional gap. The frequent edition of the kernel task is provided as follows.

$$K(U,V) = \varphi(U)^{T}\varphi(V) \tag{8}$$

In this view, the majority broadly engaged kernel tasks contain the linear kernel, Polynomial kernel, Quadratic kernel, Sigmoid and the Radial Basis task. Specified beneath are the terms for the different kernel task. For Linear Kernel:

$$linear_k(U,V) = u^T v + c \tag{9}$$

where u, v represents the inner products in linear kernel and c is a constant. For Quadratic Kernel:

$$quad_k(U,V) = 1 - \frac{u - v^2}{u - v^2 + c} \tag{10}$$

where, u, v- are the vectors of the polynomial kernel function in the input space. For Polynomial Kernel:

$$poly_k(U,V) = \left(\lambda u^T v + c\right)^e, \quad \lambda > 0 \tag{11}$$

For Sigmoid Kernel:

$$sig_k(U,V) = \tanh\left(\lambda u^T v + c\right), \quad \lambda > 0 \tag{12}$$

The effectiveness of the SVM consistently orients on the variety of the kernel. The occurrence of the attribute gap is linearly indivisible hence it has to be recorded into a better-quality dimensional gap using the Radial basis task kernel, so that the concern appears as linearly detachable. Additionally, amalgamation of any two kernel tasks is proficient to defer outstanding accuracy than that acquired by utilizing some single kernel task. In the original procedure, an original KSVM is predicted and dedicated for the noteworthy development in the categorization system. At this point, two kernel tasks such as the linear and the quadratic kernel task mutually defer outstanding presentation ratios. By uniting (6) and (7), the standard is predictable as recommended in the original technique. The mutual kernel task is successfully engaged in the KSVM and the standard of the kernel task, $avg_k(U, V)$is delivered beneath.

$$avg_k(U,V) = \frac{1}{2}\left(lin_k(U,V) + quad_k(U,V)\right) \tag{13}$$

$$avg_k(U,V) = \frac{1}{2}\left(\left(u^T + c\right) + \left(1 - \frac{u - v^2}{u - v^2 + c}\right)\right) \tag{14}$$

Error Detection victimization Kernel SVM is used to train the data that is received from the sensors, and also it detects incorrect data. This increases its responsibility and updates the training data. Error detection with new data and KSVM results are accustomed to improve the correct knowledge collection.

4.4 Error detection and correction

The main focus is on error detection for numeric big data sets from complex networks by considering specific features of the numeric data errors. The error detection process needs to filter big data sets from the network. When there is data abnormality the whole network should be traversed for finalizing the error and correcting the error. In scale free network, only few nodes in the hierarchy will have large set of links to the nodes. So based on node which has huge links can be grouped in the cluster so that the error can be located easily. So, navigate to search the error and the location of the source. The proposed clustering method can reduce the time of detection of the error and also reduce the workload of processing the whole data.

4.4.1 Error detection in sensor network

Data error detection in sensor network and complex network is unavoidable in real world complex network system. As there is a dramatic increase in big data, locating the error is also a quite challenging task with normal computing and network system.

4.4.1.1 Flat line error Flat line error (Fig. 2) indicates that nodes in the network are kept unchanged for unacceptable time series. In real world application, the transmitted data will have changes over time flow.

4.4.1.2 Data loss error Data loss error (Fig. 3) means that there is missing of data over time. It might have occurred when the data was generated or during exchange process. This requires data cleaning.

4.4.1.3 Out of bound error The value of the data is observed based on the domain knowledge gained in general. In real world applications, if any wave is beyond a fixed threshold then it is treated as out of bound error (Fig. 4).

4.4.1.4 Spike error The spike error (Fig. 5) is that in a time series, data items would be out of predicted threshold all of a sudden and normal over the time series.

4.4.2 Error correction in sensor network

The error can be corrected with the aid of correlation coefficient procedures. The calculation of the covariance (c_0) and correlation (c_r) are given in the Eqs. (15) and (16) respectively. The relationship between two random variables is measured using the covariance and correlation, after correcting a big data sensor error is discussed.

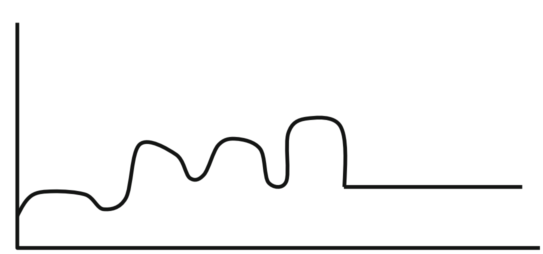

Fig. 2 Flat line error

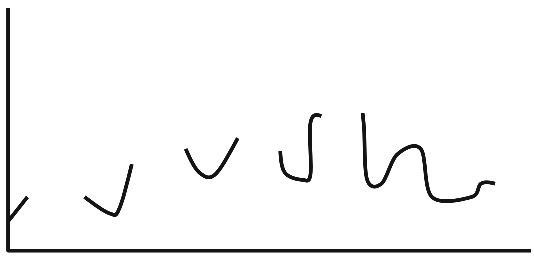

Fig. 3 Data loss error

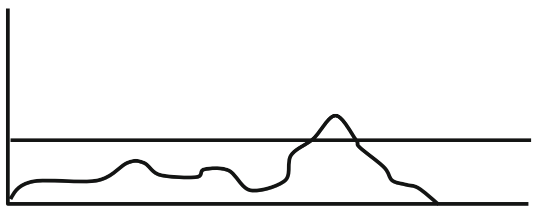

Fig. 4 Out of bound error

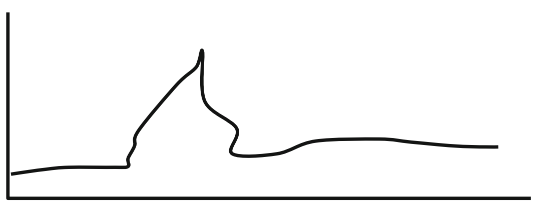

Fig. 5 Spike error

Network sensor errors are corrected with the aid of covariance correlation technique.

The following notations are used throughout, $EX = \mu x$, $EY = \mu y$, $\text{var}x = \sigma_x^2$ and $\text{var}y = \sigma_y^2$.

The covariance of X and Y is the number defined by,

$$\text{cov}(x, y) = E((x - \mu_x)(y - \eta_y)) \tag{15}$$

The correlation of X and Y is the number defined by

$$p(x, y) = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y} \tag{16}$$

Here, one exact fault rectification format cannot be implemented for the entire functions and installed circumstances of WSN. The choice of a finest fault rectification code for wireless sensor network is broadly analyzed by means of numerous researchers in the earlier period. According to its essential stage of big data set, the fault rectification is used to choose a finest appropriate fault rectification format for an exact function in wireless sensor network.

5 Results and discussion

The investigational outcome of NN-GSOFF based classifier is described below. The projected system is executed by using Java 2014 and the testing is processed with i5 processor of 3 GB RAM.

5.1 Dataset description

Linked Sensor Data [26] is an RDF dataset containing expressive descriptions of $\sim$ 20,000 weather stations in the United States. The data originated at MesoWest, a project within the Department of Meterology at the University of Utah that has been aggregating weather data since 2002.

5.2 Evaluation metrics

An approximate measurement is used to measure the efficiency of the proposed system. It comprises efficient technique that tracks the general fundamental estimation approach. Some of the measurements considered for estimation are specificity, sensitivity and accuracy.

5.2.1 Running time

On the basis of memory values, the running time in Java with Cloud Sim program is directly rooted. The time complication is usually represented in such a way that the coefficients are ignored. The time duration is computed in milliseconds (ms), in lower order terms and in most cases.

5.2.2 Memory usage

The Java with Cloud Sim program does an effective organization of the memory for use. New objects are created and placed in the stack and the memory usage of the proposed work is computed in bits.

5.2.3 Accuracy

Accuracy of the proposed method is the ratio of the total number of TP and TN to the total number of data.

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{(\text{TN} + \text{TP} + \text{FN} + \text{FP})} \tag{17}$$

5.3 Analysis process

The time based efficiency is attained from Table 1 and its corresponding graph is shown in Fig. 6a. The proposed time is measured based on a training data percentage such

Table 1 Time, memory, accuracy measure and error data on training data

Training %	Time	Memory	Accuracy	Error data
50	1,965,457	10,578,485	73.64	1268
60	2,005,484	11,184,872	74.98	1007
70	2,178,848	12,272,859	76.82	863
80	2,268,883	13,227,288	79.36	611

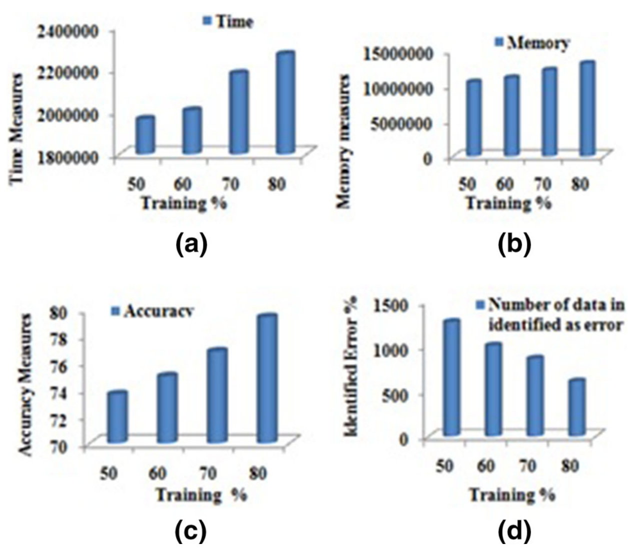

Fig. 6 Graphical representation of **a** time measures, **b** memory measure, **c** accuracy measures and **d** error data on training data

as 50, 60, 70 and 80. In the training data 50% attains a time measure of 1,965,457 ms and the remaining training set 60, 70 and 80% attains time measure of 2,005,484, 2,178,848 and 2,268,883 ms respectively. Therefore, a minimum amount of time is required to complete the process.

Table 1 also shows the cluster evaluation time measures and the performance analysis of memory by varying number of training data set percentage.

The computational memory measure is shown in Table 1 based on training set such as 50, 60, 70 and 80. A training set of 50% obtains 10578485 bits of memory. In the training set 60% attains 11184872 bits of memory and the remaining training set 70 and 80% attains a memory of 12272859 bits and 13,227,288 bits. The graphical representation is shown in Fig. 6b.

Table 1 also shows the computational accuracy measures based on training set such as 50, 60, 70 and 80%. In the training set 50% obtains 73.64% accuracy. In the training set 60% attains 74.98% accuracy and the remaining training set 70 and 80% attains an accuracy of 76.82 and 79.36% accuracy. Figure 6c shows the graphical representation of the same.

The identified errors based on training data is shown in Table 1 set as 50, 60, 70 and 80% where the identified number of error is 1268, 1007, 863 and 611 respectively. Figure 6d shows the graphical representation of the same.

5.4 Comparative analysis

The current work and the proposed work are compared in order to demonstrate that the proposed work is better. For this, FCM technique is taken to be compared with WFCM technique. The accompanying table demonstrates the similar result.

The graphical representation of the existing and proposed method is compared in Fig. 7. For clustering, Weighted FCM Clustering is used and then a Kernel SVM technique is utilized for classification. It can be claimed that our proposed work helps to attain very good results for the clustering of data. And also WFCM technique gives better outcome when compared to the Existing FCM method. The comparison outcomes are presented in the following Table 2.

From Table 2, it can be said that our proposed work gives a better accuracy. A better accuracy result is obtained on the training set of 50, 60, 70 and 80% as 73.64, 74.98, 76.82 and 79.36% from proposed WFCM whereas existing FCM gives 65.48, 68.78, 71.58 and 75.89% of accuracies.

Table 3 shows the comparison results of proposed KSVM and existing SVM. Here 80% of the training data is taken and the accuracy for proposed KSVM is greater than existing SVM which is 85.56% and 82.41% respectively. The proposed KSVM gives an accuracy of 80.03% in neural network and accuracy of 79.12% in Fuzzy network. From Table 3, the Kernel based SVM outperforms the existing methods because of effective clustering using WFCM.

Hence, our proposed WFCM clustering and KSVM produces high accuracy. Hence, the proposed WFCM accuracy and KSVM accuracy is better than the existing techniques (Fig. 7).

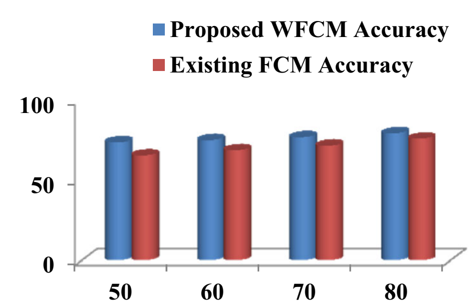

Fig. 7 Comparison of proposed and existing method

Table 2 Comparison of proposed and existing clustering accuracy

Training %	Proposed WFCM accuracy	Existing FCM accuracy
50	73.64	65.48
60	74.98	68.78
70	76.82	71.58
80	79.36	75.89

Table 3 Comparison of proposed KSVM and existing SVM

Methods	Accuracy (in %)
Proposed KSVM	85.56
SVM	82.41
Neural network	80.03
Fuzzy	79.12

6 Conclusion

Various implementation strategies of error detection techniques in WSN are presented. The main aim is at particular implementation strategy for error detection and correction is discussed with the aid of Weighted FCM and KSVM. In order to detect and find the location of error in big data set, a sensor network system is mainly used and a novel approach is developed with cloud computing. First, the classification of error in big data sets is presented. Second, the correlation comparison between sensor network systems and the scale-free complex networks are introduced. Accordingly the error types are defined. Different strategies for detecting and locating errors in big data sets on cloud are used.

Acknowledgements R. Sheeba gratefully thanks UGC fellowship provided by Government of India, NewDelhi, India. The scholarship id is 201516-MANF-2015-17-TAM-54669.

References

1. Muhammed, T., Shaikh, R.A.: An analysis of fault detection strategies in wireless sensor networks. Proc. J. Netw. Comput. Appl. **78**, 267–287 (2017)
2. Yang, C., Chang, L., Xuyun, Z., Surya, N., Jinjun, C.: A time efficient approach for detecting errors in big sensor data on cloud. Proc. J IEEE Trans. Parallel Distrib. Syst **26**(2), 329–339 (2015)
3. Kumar, A.R., Ashok, M., Sam, R.P.: Error detection and cleaning for big data sets from sensor network systems on cloud. Int. J. Comput. Sci. Trends Technol. **4**(5), 261–265 (2016)
4. Sreenivasulu, P., Rudra Kumar, M., Subramanyam, A.: Data error detection and time reduction for big sensor data sets. In: Proceedings of International Journal of Computer Science and Engineering and Scientific Technology (2016)
5. Damini, V.S., Priyashree, K., Jayashubhaj, K.: Optimized error detection analytics with big data on cloud. Proc. Int. J. Innov. Res. Sci. Eng. Technol. **5**(10), 236–242 (2016)
6. Bhuvaneswari, S., Sibiya, I.: Temporal approach for big data error detection and correction on cloud in WSN. Proc. Int. J. Adv. Res. Trends Eng. Technol. **3**(2), 259–264 (2016)
7. Jyothsna, C., Suhasini, N.S.: An approach for detecting and analyzing errors of big sensor data on cloud. Proc. Int. J. Adv. Technol. Innov. Res. **8**(10), 1946–1950 (2016)
8. Patil, D.S., Gaikwad, R.: An efficient approach to error detection and data recovery using map reduce. In: Proceeding of International Conference on In Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 35–39 (2016)
9. Goyal, Ankur, Nareda, Vinita: A review studies on burst error detection and correction in big data. Proc. Int. J. Sci. Res. **6**(5), 943–947 (2017)
10. Anusha, G, Guruprasad, N.: Reducing the time factor for analyzing the errors and fixing in big sensor data. In: Proceedings of International Journal of Advanced Networking & Applications (IJANA), pp. 78–81, Special issue
11. Neela, S., Chandramouli, H., PrasadBabu, B.R.: An approach for detecting and correcting errors of big sensor data. Proc. Int. J. Eng. Res. **5**(4), 790–991 (2016)
12. Goyal, A., Nareda, V.: A fast block-level error identification and rectification of error burst in big data on cloud. Proc. Int. J. Adv. Res. Comput. Sci. **8**(5), 2486–2490 (2017)
13. Pansare, J.R., Bajad, V.D.: Errors detection in big sensor data on cloud using time efficient technique. In: Proceedings of the ACM Symposium on Women in Research 2016, pp. 12–14
14. Logapriya, R., Preethi, J.: Efficient methods in wireless sensor network for error detection, correction and recovery of data. Proc. Int. J. Novel Res. Comput. Sci. Softw. Eng. **3**(2), 47–54 (2016)
15. Sabde, P., Shahapure, P., Sable, S., Pagar, P., Patil, V.: Detection and recovery of errors on cloud in efficient time using hadoop. Proc. Imp. J. Interdiscip. Res. **2**(6) (2016)
16. Oliveira, J.C.M., Pontes, K.V., Sartori, I., Embiruçu, M.: Fault detection and diagnosis in dynamic systems using weightless neural networks. Proc. J. Expert Syst. Appl. **84**, 200–219 (2017)
17. Cai, B., Liu, H., Xie, M.: A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks. Proc. J. Mech. Syst. Signal Process. **80**, 31–44 (2016)
18. Lo, C., Lynch, L.P., Liu, M.: Distributed model-based nonlinear sensor fault diagnosis in wireless sensor networks. Proc. J. Mech. Syst. Signal Process. **66**, 470–484 (2016)
19. Feng, G., Zhao, H., Gu, F., Needham, P., Ball, A.D.: Efficient implementation of envelope analysis on resources limited wireless sensor nodes for accurate bearing fault diagnosis. Proc. J. Meas. **110**, 307–318 (2017)
20. Guesmi, H., Salem, S.B., Bacha, K.: Smart wireless sensor networks for online faults diagnosis in induction machine. Proc. J. Comput. Electr. Eng. **41**, 226–239 (2015)
21. Suresh, A., Varatharajan, R.: Competent resource provisioning and distribution techniques for cloud computing environment. Clust. Comput. (2017). https://doi.org/10.1007/s10586-017-1293-6
22. Panda, M., Khilar, P.M.: Distributed Byzantine fault detection technique in wireless sensor networks based on hypothesis testing. Proc. J. Comput. Electr. Eng. **48**, 270–285 (2015)
23. Zhiyang, Y., Zhao, X., Wan, H., et al.: A novel fault diagnosis mechanism for wireless sensor networks. Math. Comput. Model. **54**(1), 330–343 (2011)
24. Alic, A.S., Tomas, A., Medina, I., Blanquer, I.: Muffinec: error correction for de novo assembly via greedy partitioning and sequence alignment. Elsevier Inf. Sci. **329**, 206–219 (2016)
25. Weng, Y., Ilić, M.D., Li, Q., Negi, R.: Distributed algorithms for convexified bad data and topology error detection and

identification problems. Elsevier Int J Electr Power Energy Syst **83**, 241–250 (2016)
26. Linked sensor data from http://wiki.knoesis.org/index.php/LinkedSensorData

R. Sheeba obtained her Bachelor's degree in Information Technology from St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil and Master degree in Information Technology from SNS college of Engineering and Technology, Coimbatore. Currently, she is doing research in the area of big data management system in PSN College of Engineering and Technology under Anna University, Chennai.

G. Jiji received her B.E. degree in Electronics and Communication Engineering from St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, M.Tech. degree from Sastra University in power electronics and drives and Ph.D. degree in Electrical Engineering from Anna University, Chennai. She has been currently employed as Associate Professor in the Department of Electronics and Communication Engineering at PSN college of Engineering and Technology, Tirunelveli. Her research interests include wireless sensor networks, Control Systems, Power converters and Internet Programming.

T. Ajith Bosco Raj received his B.E. degree in Electronics and Communication Engineering from St. Xavier's Catholic College of Engineering, Chunkankadai, Nagercoil, M.E. degree in control and instrumentation from St. Xavier's Catholic College of Engineering and Ph.D. degree in Electrical Engineering from Anna University, Chennai. Presently he has been employed as the Professor and Head of the Department of Electronics and Communication Engineering at PSN college of Engineering and Technology, Tirunelveli. His research interests include wireless sensor networks, Control Systems, power converters, embedded systems design and robotics.