[1]Ananda Ravuri,
[2]Dr. R. Josphineleela,
[3]Dr. G. V. Sam Kumar,
[4]Kiranmai R,
[5]Dr. Thangiah SathishKumar,
[6]Dr. A. Rajesh Kumar

Machine Learning-based Distributed Big data Analytics Framework for IoT Applications

Journal of Electrical Systems

Abstract: - The emerging trend of the Internet of Things (IoT) leads to more real-time applications and it raises the accumulation of more structured and unstructured data. The processing of unstructured and structured data for different distributed applications becomes arduous. Moreover, the accuracy, load balancing, and latency of the services are also challenging. Some of the state-of-art works failed to achieve those parameters. In context with these, we proposed a machine learning-based novel approach that utilizes an SVM classifier. The SVM classifier can be used for the classification and data analytic purposes of features extracted from the data processing step. Data processing employs two steps such as data extraction and data scaling. The data extraction is performed by the adoption of the Principal Component Discriminant power-based Linear Discriminant analysis (PCDP-LDA) technique. The big data framework of the proposed model is evaluated using a tool called Weka. Meanwhile, the data scaling is performed by the Naïve Bayes approach and divides the extracted features into blocks. Experimental analysis is performed and compared with existing approaches. Our proposed approach provides more effective classification and data analytic accuracy than the other approaches. Our approach also provides better latency, and load balancing of data in the distributed big data analysis.

Keywords: *Machine learning, SVM, feature extraction, device layer, data scaling, and data analytics*

1. Introduction

Mostly, the Internet of Things (IoT) is enchanting with data networks that link with devices over the transmitted networks [1]. In our day-to-day life, IoT plays the main role in many smart things. It helps humans to make their job easier and quicker. Mobiles, sensors, and other household devices are helpful for humans in their everyday life. In an IoT system, the interactions between the devices are set up to trade [2, 3]. The data is complex to analyze and manage from a large amount of data. It can detect an image based on this technology. A detailed analysis of cloud images can be acquired by the Internet of Things. The functions between IoT and data are modules by cloud processing. While placing the cloud the information is verified and compared with IoT devices. In between IoT and extreme data, the cloud is considered the connecting agent. If there is no information in IoT devices the information is obtained by comparing the images. Cloud processing and module are the core functions of IoT devices [4].

Further, the feature acquisition module mentions the image detection and cloud processing module. The feature acquisition module is utilized by the image acquisition module. The image features can be collected and analyzed by the feature module. IoT devices provide higher efficiency in cloud computing. Load sharing and recomputed data are the issues found in cloud processing. In IoT, edge computing plays a main role in applications [5]. Data combination is intended to provide a sensor output description. In real-time environment data from several sources are derived for the decision-making process. Data combination is supported in the decision-making process to assist the data exploration process. Decision-making is the process of making decisions at the

[1]Senior Software Engineer Intel corporationHillsboro, Oregon 97124 USA
Ananda.ravuri@intel.com
[2]Professor, Department of computer science and Engineering,Panimalar engineering college
pecleela2005@gmail.com
[3]Associate Professor, Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India. gvsamkumar@gmail.com
[4]Assistant Professor,Department of CSE(Data Science),B V Raju Institute of Technology,Narsapur, TS, India
kiranmai.r@bvrit.ac.in
[5]Associate Professor,Department of Computer Science and Engineering,
Hyderabad Institute of Technology and Management, Hyderabad
tsathish.jobs@gmail.com
[6]Associate Professor, Computer Science and Engineering, N.S.N. College of Engineering and Technology,
MANALMEDU, Karur 639003, Tamilnadu, India.
arurajesh1980@gmail.com

proper time. It can provide suitable information. The data network is reduced over the network to avoid improper data. The volume of data is proportional to the transmitted data possibility of occurring errors in higher data [6, 7].

Due to their capacity to represent and quantify aspects of uncertainty, fuzzy sets have been used for big data processing. Fuzzy set methods and tools, such as fuzzy system extenders and generalizations, fuzzy logic, and fuzzy systems, have emerged as intriguing and practical tools for GRC [23]. Fuzzy sets have been employed in a variety of fields, including control systems, pattern recognition, and machine learning. We can represent and process information using fuzzy sets at various information finer-grained levels. Numerous efforts have been taken up to this point that center on using fuzzy sets to handle and/or comprehend big data. In the context of big data, fuzzy set techniques appear promising or have already shown some benefits for at least four main reasons.

The cost of the sensors and selectors is low most organizations are focused on this technology. Due to high speed, the volume and data are large. It helps to improve decision making which is related to smart appliances. The issues that are hiked by IoT are due to the increase in smart applications. A large amount of data can be detached from IoT data for decision-making to extract patterns. Scalability, bandwidth, and resource management are issues that can be rectified by future IoT devices [8]. The existing approaches failed to acquire effective latency, computation efficacy, and classification of IoT data features. To overcome these issues we proposed a novel machine learning approach that adopted the SVM for the effective classification of data and big data analysis. The contributions of the work are enclosed below,

Moreover, the accuracy, load balancing, and latency of the services are also challenging. Some of the state-of-art works failed to achieve those parameters. In context with these, we proposed a machine learning-based novel approach that utilizes an SVM classifier. The SVM classifier can be used for the classification and data analytic purposes of features extracted from the data processing step. Data processing employs two steps such as data extraction and data scaling. The data extraction is performed by the adoption of the Principal Component Discriminant power-based Linear Discriminant analysis (PCDP-LDA) technique. The big data framework of the proposed model is evaluated using a tool called Weka. Meanwhile, the data scaling is performed by the Naïve Bayes approach and divides the extracted features into blocks. Experimental analysis is performed and compared with existing approaches. Our proposed approach provides more effective classification and data analytic accuracy than the other approaches. Our approach also provides better latency, and load balancing of data in the distributed big data analysis.

- In this manuscript, Distributed Bigdata analysis based on Support Vector Machine based approach is proposed.
- The datasets are acquired from the IoT nodes. Data processing employs two steps such as data extraction and data scaling.
- The data extraction is performed by the adoption of the Principal Component Discriminant power-based Linear Discriminant analysis (PCDP-LDA) technique.
- Meanwhile, the data scaling is performed by the Naïve Bayes approach and divides the extracted features into blocks.
- Subsequently, the processed data are forwarded to the SVM-based classifier for classification and analysis purposes.
- The big data framework of the proposed model is evaluated using a tool called Weka.
- The proposed approach provides more effective classification and data analytic accuracy than the other approaches such as Fuzzy, DPDCM, and DCCM. This approach also provides better latency, and load balancing of data in the distributed big data analysis.

The rest of the article is arranged as in section 2 the relevant works are analyzed and reviewed. Section 3 illustrates the proposed processes such as data gathering, data processing, and data classification and analysis in a clear manner. The experimental analysis and discussion are made in section 4. The work is concluded n section 5.

2. Literature survey

Li et al. [9] suggested a deep convolutional computation model (DCCM). It learns the hierarchy characteristics of large datasets by extending the CNN from vectors to the conical area using the tensor representation paradigm. To avoid over-fitting and increase learning happens, a scalar inversion procedure is devised to create complete use of the interest points and morphologies present in the huge data. An elevated back propagation neural approach is also developed for training the deep convolutional computing model's variables in high-order fields. Furthermore, trials on test datasets, namely STL-10, SNAE2, and CUAVE, are conducted to validate the DCCM's effectiveness. The deep convolutional computing paradigm can provide better classification accuracy for huge data in IoT to the multimodal model, according to experimental results.

Li et al. [10] introduced a deep learning (DL) model. DL is a potential method for obtaining precise features from various sensor data from IoT devices in challenging situations. Deep learning is also suitable for the edge computing environment due to its layered structure. As a result, we'll start with an introduction to deep learning for IoTs in the edge computing environment in this post. They build a new unloading technique to optimize the performance of IoT deep learning applications with edge devices because current edge devices offer restricted processing capacity. In the performance review, the DL technique to test by running various deep-learning tasks in an edge computer system. The testing findings suggest that our strategy beats previous machine learning for IoT optimization methods.

Hossain et al. [11] introduced a machine learning (ML) method to provide surgical outcome prediction in total knee arthroplasty. The challenge was handled by using general linear regression (GLR) analysis to build forecasting analytics from the leg kinematic information of 35 osteoarthritis patients who had a posterior stabilized implant. Two prediction approaches were proposed, as well as their sub-classes, and they were then assessed using a leave-one-out cross-validation procedure. With a Pearson's correlation coefficient (cc) of 0.840.15 (mean SD) and an RMSE of 3.271.42 mm for frontal vs. flexor muscles and cc of 0.890.15 and RMSE of 4.251.92° with extension, the optimum technique (i-e pattern). Though those are verified for one kind of prosthetic, the criteria might be applied to certain other devices.

A double projection model with deep computation (DPDCM) was described by Zhang et al. [12]. They also devised a learning strategy for training the DPDCM. Cloud hosting would be used to enhance overall subsequent revisions of the training algorithm with the data on cloud crowd-sourcing. A privacy-preserving DPDCM based on the BGV encrypting technique is presented to secure private data. Furthermore, investigations on NUS-WIDE-14 and Animal-20 are done to compare the efficacy of PPDPDCM and DPDCM. DPDCM provides greater accuracy of classification than DCM, according to the findings. Most significantly, PPDPDCM may successfully enhance train variable effectiveness, demonstrating its potential for massive data-extracting features.

Jeong et al. [13] presented the advantages of big data analysis. They cover a wide range of topics, including software engineering, decision-making, the semantic web, encrypted data query processing, intrusion detection methods, malware distribution networks, human tracking techniques, fingerprint matching, and image segmentation.

In 2020, Sankaranarayanan et.al [14] presented an integration of data flow and distributed deep learning in the IoT-Edge environment to bring down the latency and increase accuracy starting from the data generation phase. To this end, a novel Data Flow and Distributed Deep Neural Network (DF-DDNN) based IoT-Edge model for big data environment has been proposed. This method has resulted in a latency reduction of up to 33% when compared to the existing traditional IoT-Cloud model.

In 2021, Fang et.al [15] presented an efficient FL framework called FL-PQSU. It is composed of a 3-stage pipeline: structured pruning, weight quantization, and selective updating, which work together to reduce the costs of computation, storage, and communication to accelerate the FL training process. This study FL-PQSU using popular DNN models (AlexNet, VGG16) and publicly available datasets (MNIST, CIFAR10), and demonstrate that it can well control the training overhead while still guaranteeing the learning performance.

3. Proposed structure

The offered machine learning techniques, such as the SVM approach, are employed due to factors like energy usage, data accessibility, and scalability. In IoT contexts, where data streams are continuously generated from various connected devices, the volume of data can be enormous, making it difficult to deploy DNNs, which frequently require big datasets for training. SVM, on the other hand, is a more sensible option for IoT-based applications due to its reputation for effectively handling high-dimensional data and adaptation for remote computing. Additionally, energy use is a major issue in IoT installations because devices may only have a limited amount of power. In comparison to SVM, which has relatively low computational demands, DNNs frequently need a lot of processing power, which reduces their energy efficiency. The authors may have chosen SVM-based techniques in this specific setting in order to strike a balance between energy efficiency, data scalability, and acceptable performance for Distributed Bigdata analysis in IoT-based applications, even if DNNs can offer outstanding accuracy in some cases.

Modern smart devices are connected by a framework known as IoT and thus collect an enormous amount of data. This generates bigdata between the IoT application-based devices and sensors. The data gathering has been done in the device layer [16]. The data processing layer significantly makes the feature extraction and scaling processes. The rise in the production of data became crucial in acquiring latency, accuracy, and trust. Thus data analytics become a challenging one. To overcome this challenge we have utilized next-generation-based functionalities for both structured and unstructured data. The feature extraction is carried out by the PCDP-LDA approach and the scaling is performed by Naïve Bayes. The computational power of the IoT devices is obtained by the inclusion of base stations, networking devices, and machine learning at the edge level. The extracted data are forwarded promptly to the cloud layer with the employment of base stations and networking devices. This circumvents the problems like load balancing and energy efficiency.

Meanwhile, the accuracy, storage, and speed are meager due to the single data center in the cloud storage layer. To tackle these issues, we have proposed an SVM-based classifier that helps to analyze the data itself. Moreover, this enhances the parameters mentioned above in the IoT network. The schematic overlay of the proposed work is elucidated in Figure 1.

3.1 Main elements of proposed SVM based distributed bigdata analysis

The main elements of the proposed approach are portrayed in this section. This includes three steps such as (i) data collection, (ii) data processing, and (iii) data analysis. The first step is performed in the IoT layer, the second one is evaluated in the edge layer, and the final step is conducted in the cloud layer. The flow diagram of our proposed work is illustrated in Figure 2.

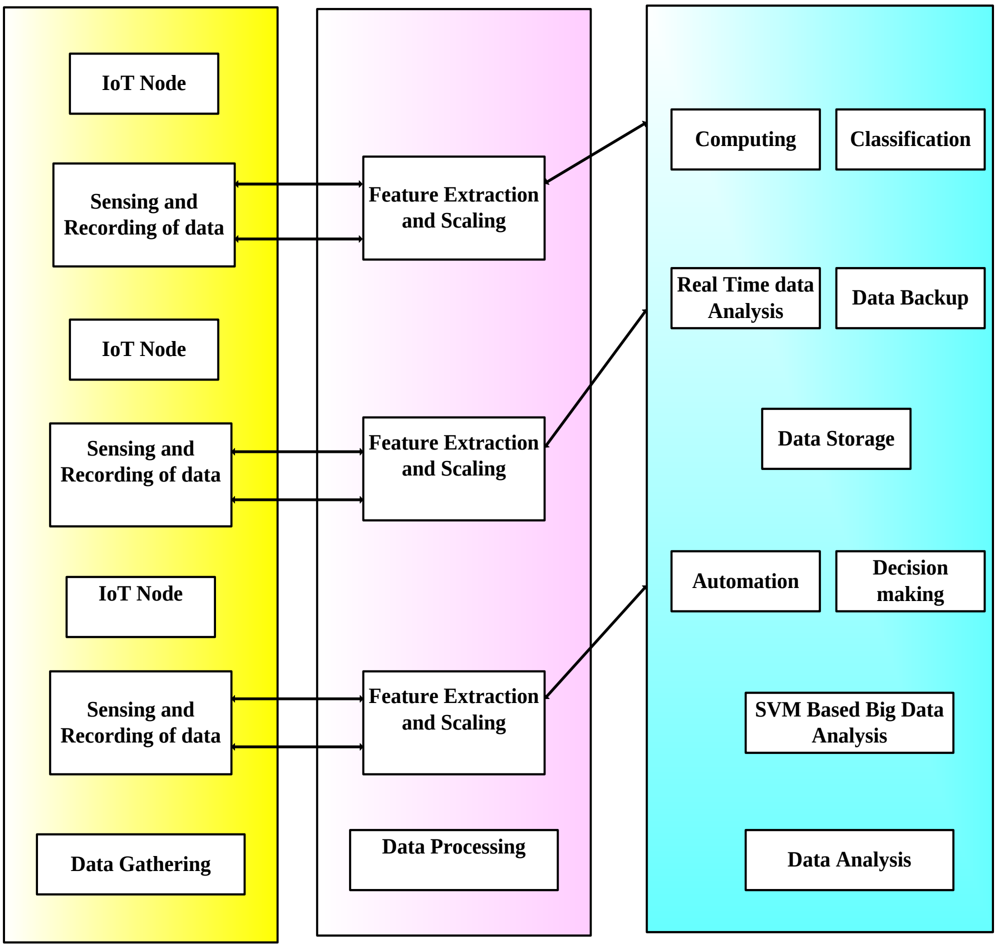
IoT Node
Sensing and Recording of data
IoT Node
Sensing and Recording of data
IoT Node
Sensing and Recording of data
Data Gathering
Feature Extraction and Scaling
Feature Extraction and Scaling
Feature Extraction and Scaling
Data Processing
Computing
Classification
Real Time data Analysis
Data Backup
Data Storage
Automation
Decision making
SVM Based Big Data Analysis
Data Analysis

Fig 1: Schematic Overlay of the proposed approach

3.2 Data collection

The data from the various IoT devices are collected in the device layer. The data includes multimedia, video, audio, and more. In real-time, the data are sensed by the sensor and collected by the IoT devices. The device layer also encloses the data such as position, status data, automation, etc., subsequently, the internal sensors gather information from the smart devices such as health care monitors, smart television, mobiles, and various commercial devices (security, forecasting) [17].

3.3 Data Processing

This is performed in the edge layer and is an essential element in the data analysis system. Boundless data generated from the IoT devices are processed to convert them into a required format for further action. The data processing includes PCDP-LDA-based feature extraction and Naïve Bayes-based data scaling.

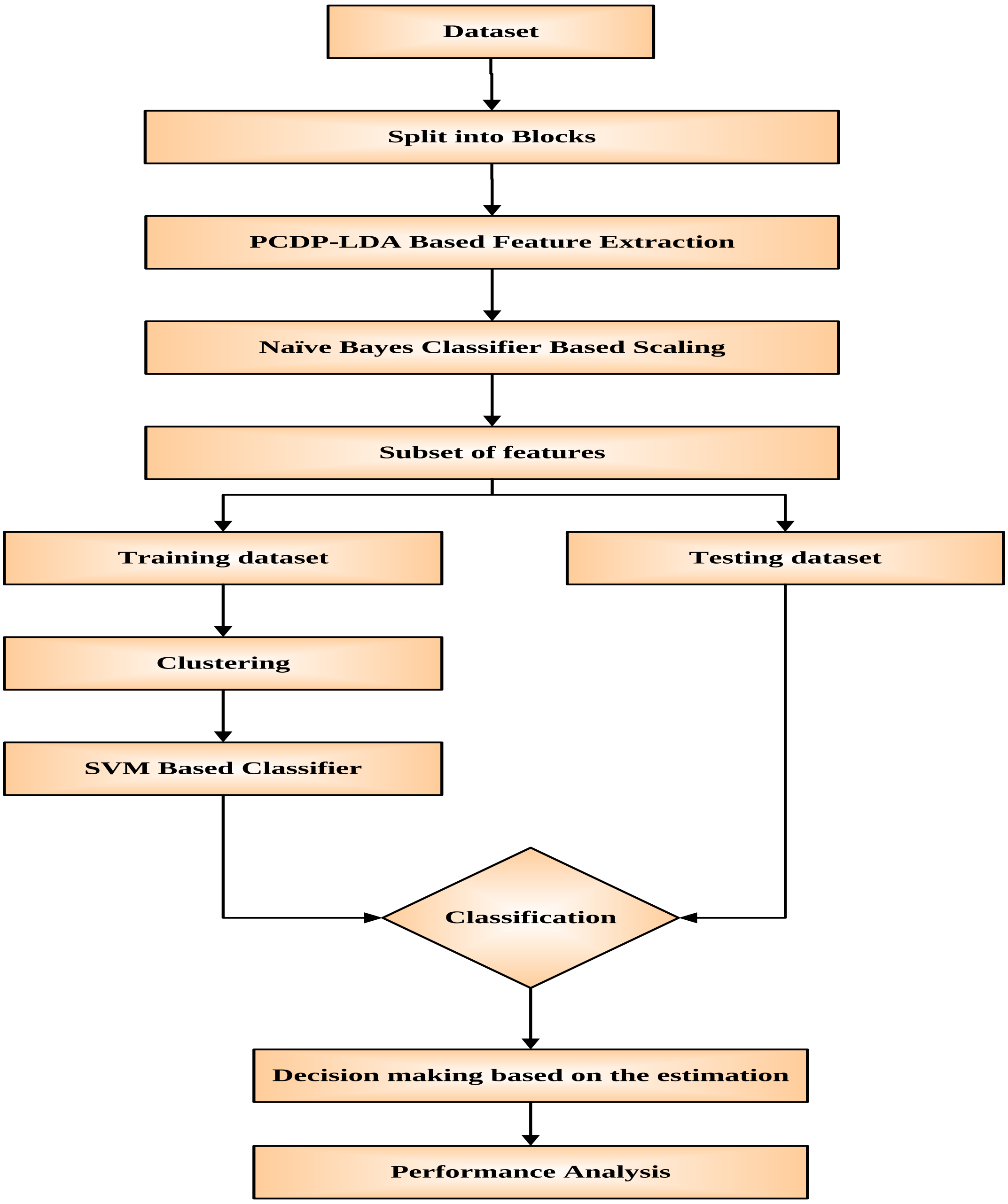

Fig 2: Flow sketch of proposed work

3.4 Extraction of features

This is mainly performed to identify the relevant and irrelevant features with the labels in the IoT applications. The load balancing, communication bandwidth enhancement, and mitigation of time delay are performed by this method. Since the data are continuously increasing and therefore the data are categorized as usage of data, quality of data, and big data. Data usage includes noiselessness, completeness, and semantics. Meanwhile, the quality of data means redundancy, load balancing, efficiency, and accuracy. Big data is further grouped as velocity, veracity, volume, and variety. Further, the components and communication protocols of the processing system are categorized as (i) device to server, (ii) server to server, and (iii) device to device. For the feature extraction, we have utilized the PCDP-LDA [18] approach. This process will ignore the noises and redundant features and thus help us to acquire relevant and useful information. These extracted features can be used for the identification of necessary information in the IoT-based machine-learning environment.

Prior to the feature extraction, the IoT data are split into different blocks which might be fixed or variable. The scaling and feature extraction is performed in the blocks. Meanwhile, training sets include sampled inputs that are exploited by machine learning. The input vector and its respective output vectors (labels) are utilized by the supervised learning concept. For unsupervised learning, the vectors are not needed and the reinforcement learning approach is used for the particular scenario. However, the transformation of features into variable samples or space enhances the outcomes, and this process is termed feature extraction.

The collected data are sent to Principal Component Discriminant power-based Linear Discriminant analysis (PCDP-LDA) [19]. The main aim of the PCDP-LDA is that it is used to overcome overfitting issues and mitigate the dimensionality of the collected data. Since it possesses several advantages we are using it for our proposed framework. The steps involved in the proposed algorithm are listed below,

- The maximum number of principal components m can be defined in terms of the training matrix $Z_{train}(a \times b)$. That can be determined as $(a \geq b)$ for$m = (a-1)$.
- The training matrix $Z_{train}$is decomposed with the principal components m. Then the transpose of the loading vector L_{train}^{T}can be multiplied with the score vector V_{train} to acquire the training matrix.

$$Z_{train} = [L_{train}^{T} V_{train}]_1 + [L_{train}^{T} V_{train}]_2 + \ldots.., + [L_{train}^{T} V_{train}]_k + H_{train} \tag{1}$$

- Then the test set T_{test} can be evaluated from the training set Z_{train}. It is expressed as,

$$T_{test} = L_{train}^{T} Z_{test} \tag{2}$$

- With respect to the score vector $SV_j$and the discriminability g_j the principal component can be evaluated. where

$$g_j = \frac{C_j}{CW_j} \tag{3}$$

Here C_j and CW_j are the class dispersed between and within score vectors. The within-class dispersion can be evaluated by the

$$CW_j = \sum_{i=1}^{l} C_{ij} \tag{4}$$

Here, the *l* represents the number of classes and the class dispersion between the score vectors is estimated as,

$$C_{ij} = \sum_{m \in L_i} \left[z_j^m - a_{ij} \right]^2 \tag{5}$$

The mean value of the score vector is denoted as a_{ij}in mth features and then the score value can be indicated as t_j^m. Then the mean value of the score vector is determined as,

$$b_{ij} = \frac{1}{d_i} \sum_{m \in L_i} t_j^m \tag{6}$$

Henceforth, the class dispersion is estimated as

$$CD_j = \sum_{i=1}^{l} b_i \left[a_{ij} - a_j \right]^2 \tag{7}$$

Here, a$_j$ is the mean training feature value. From the above, we can conclude that the optimal feature extraction is obtained by the leave-one-cross-validation (LOOCV) value by the exploitation of the training dataset score. Moreover, the construction of Z_{train}can is acquired by the LDA method which can be used to obtain the predicted features. The proposed PCDP-LDA enhances the feature extraction due to its unique features like self adaptability and robustness.

3.5 Scaling of data

It is the process of splitting the dataset into valued groups. The groups are formed by categorizing the same valued features. One feature from one group must be different than the feature from another group. The purpose of scaling the data is to arrange the data with similar properties in the same group. We have adopted the Naïve Bayes algorithm [20] for this process.

Naïve Bayes (NB) is one of the widely used classifiers in various domains such as medical diagnosis, pattern and image recognition, bioinformatics, and weather forecasting. Both equally and independently feature classification decisions are contributed via NB [21]. Based on real-world conditions, the NB is insufficient and it increases the computational complexity.

Let us assume, the target class TC is$TC = \{tc_1, tc_2, tc_3, \ldots, tc_n\}$with the new item vector (*IV*) and it considers the feature vectors$FV = \{fv_1, fv_2, fv_3, \ldots, fv_n\}$.

$$T\arg et_{IV} = \underset{e_j \in TC}{\arg\max}\left[R\left(d_j \middle| FV\right)\right] \tag{8}$$

From the above equation, the given feature vector is $FV$and the class $d_j$conditionally probability is$R\left(d_j \middle| FV\right)$. The below equation expresses the independence of these features.

$$Pr\ o\ (FV) = Pr\ o\ (fv_1, fv_2, fv_3, \ldots, fv_n)$$
$$= Pr\ o\ (fv_1) \times Pr\ o\ (fv_2) \times Pr\ o\ (fv_3), \ldots, Pr\ o\ (fv_{m1})$$
$$= \prod_{j=1}^{m} Pr\ o\ b\left(fv_j\right) \tag{9}$$

The target class is defined as,

$$T\arg et_{IV} = \underset{e_j \in TC}{\arg\max}\left[\Pr o(d_j) \times \prod_{j=1}^{m} R\left(fv_j \middle| d_j\right)\right] \tag{10}$$

Depending upon the real-world applications, the features consider and assign an equal weight value.

$$T\arg et_{IV} = \underset{e_j \in TC}{\arg\max}\left[\Pr o(d_j) \times \prod_{W_j \in R} \Pr o\left(fv_j \middle| d_j\right)^{W_j}\right] \tag{11}$$

Each weight and feature areW_j and FV_j. The significant features are represented based on the positive number. From another point of view, promoting the performance of the WNB classifier can be achieved by compensating its performance with another heuristic besides conditional and prior probabilities.

3.6 Big data analysis

This is the final stage of the work is classification and big data analysis. To perform this we utilized a machine-learning approach known as an SVM classifier [22]. This can be utilized to overcome the issues like data handling problems and address centralization issues. this can also be used to reduce energy consumption and excessive data in IoT applications. The main objection of the SVM classifier is that effectively classifies the features that are obtained from the data processing process by considering the hyper-plane of the data. The formulation of hyperplane is defined as,

$$Vp(a) = \omega^T a + xh = 0 \tag{12}$$

The input is represented as a, and its respective bias value is given as xh, and the weight vector can be indicated as ω. Along with the hard margin optimality [23], the classes of training datasets are perfectly separated. The distance between the hyperplane and adjacent training feature points is increased with the appropriate selection of hyperplane decision constraints.

Meanwhile while considering the nonlinear classification the hyperplane will lie in the feature point other than the original input by holding the constraint. Then the equation (12) can be modified as,

$$\omega^T \theta(a) + xh = 0 \tag{13}$$

The transformation of the input vector into nonlinear transformation [24] can be achieved by the $\theta(a)$. The weight of the vector can be optimized with the exploitation of the Lagrange multiplier as shown in equation (14).

$$\omega = \sum_{i=1}^{M} \rho_i b_i \theta(a) \tag{14}$$

Here ρ_i denotes the Lagrange multiplier's coefficients. Then the optimized decision can be obtained as shown below,

$$\sum_{i=1}^{M} \rho_i b_i \theta(a_i) + xh = 0 \tag{15}$$

Moreover, let us consider $r_i = \rho_i b_i$and the $\kappa(a, a_i) = \theta(a)^T \theta(a_i)$ and makes a decision as shown below,

$$y = \sum_{i=1}^{M} r_i \kappa(a, a_i) + xh \tag{16}$$

For decision making the SVM follows the steps illustrated in figure 3. The output is represented by b and if its value is -1 then it represented class -1 and vice versa.

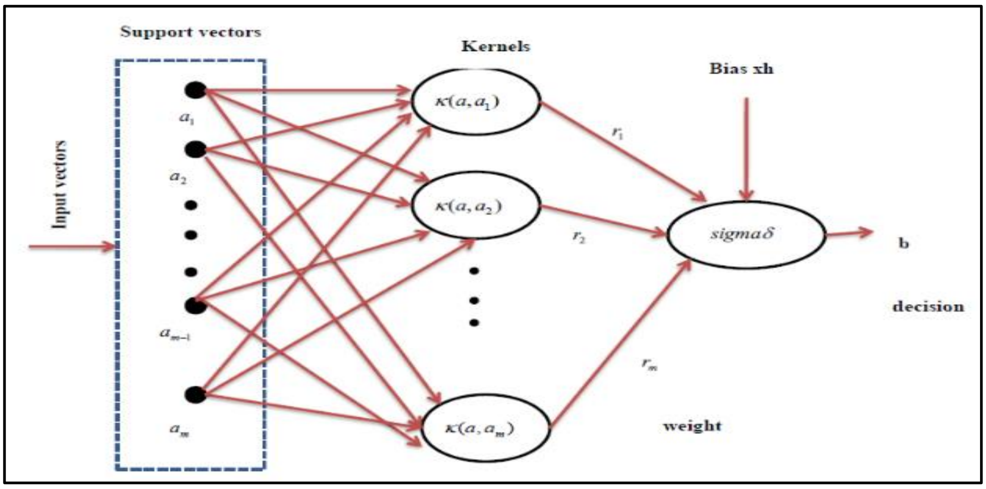

Fig 3: Big data analysis by using the SVM classifier

4. Result and Discussion:

The experimental investigations of the proposed methodology are detailed and described in this section. The state-of-art of comparative results are plotted and analyzed. The big data framework of the proposed model is evaluated using a tool called Weka. The parameter configuration of the proposed model is given in Table 1. Table 2 expresses the classifier being used to classify and analyze data that has been collected during the information processing stage.

Table 1: Parameter configuration

Parameters	Ranges
Population size	50
Maximum number of iterations	100
Kernel function	Gaussian
Kernel scale	0.56

Table 2: IoT applications based on the big data analysis classes

Real_Time	Green plum	HANA	Parallel processing	Memory based
Offline	Skribe	Kafka	Tume Tunnel	Chukwa
Database level	MangoDB	TB Level Data		
Manufacturing level	Data analysis plan	Distributed file	TB-level data	-
Large level	MapReduce	Scala	-	-

4.2. Dataset description:

The detailed dataset description is plotted in this section. The research offered a new computer studying strategy that makes use of a Classification model. Raw data and data scaling are two processes in the data processing process [25-29]. The Principal Component Discriminant power-based Linear Discriminant analysis (PCDP-LDA) method is analyzed. Furthermore, the Naive Bayes technique is used to scale the information and partition the collected features into blocks. An experimental analysis is carried out and compared to existing methods [30-33]. Compared to existing methodologies, the suggested technique delivers excellent categorization and computational intelligence reliability. In distributed big data analysis, our technique also delivers superior data load balancing and latency.

4.3. Performance matrices:

Accuracy, specificity, sensitivity, precision, recall and F-measures are the performance measures [35-37]. Each of these measures are described and explained as follows;

$$Accuracy = \frac{T_P+T_N}{T_P+T_N+F_P+F_N} \qquad (17)$$

$$Sensitivity = \frac{T_P}{T_P+F_N} \qquad (18)$$

$$Re\,c\,all = \frac{T_P}{T_P+F_N} \qquad (19)$$

$$Pr\,e\,cision = \frac{T_P}{T_P+F_P} \qquad (20)$$

$$Specificity = \frac{T_N}{T_N+F_P} \qquad (21)$$

$$F - measure = 2 \times \frac{Re\,call * Pr\,ecision}{Re\,call + Pr\,ecision} \qquad (22)$$

Based on the above equations, the truly positive and the truly negative classes areT_P and T_N. Where, $F_P$and $F_N$are the false positive and false negative classes.

4.4. Performance Analysis:

Fig 4 expresses the performance analysis of latency. Here, we have used MS-COCOC, MNIST, and KDD Tests is the three commonly used datasets. From this, the latency is calculated here in which the unit of latency is measured in seconds. From this plot, we have obtained 0.11 sec, 0.57 sec, and 0.048 sec latency results by using the datasets like MS-COCOC, MNIST, and KDD Test respectively.

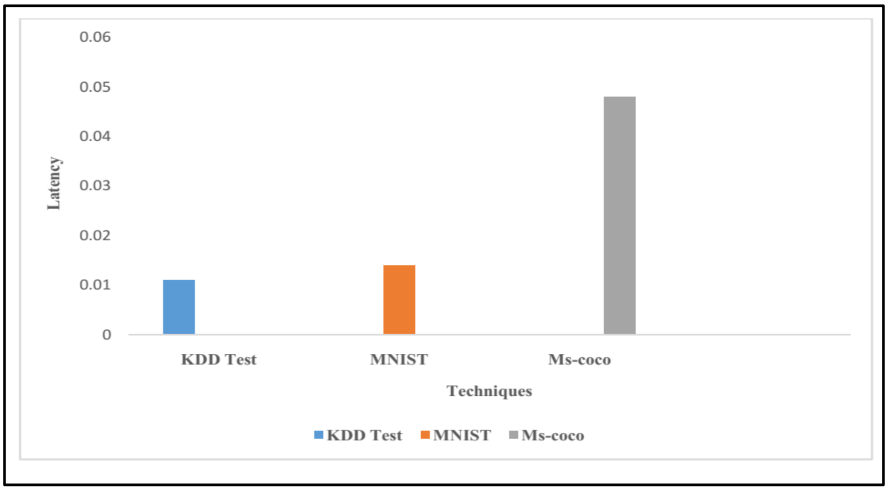

Fig 4: Performance analysis of Latency

The state-of-art result of accuracy is plotted in Fig 5. The state-of-art methods like Fuzzy, DPDCM, DCCM, and the proposed method evaluate the performance of accuracy, which is measured in percentage. The state-of-art techniques like Fuzzy, DPDCM, DCCM, and the proposed method yielded 40%, 72%, 50%, and 89% accuracy results. From this investigation, the proposed method offers superior classification accuracy than other existing methods.

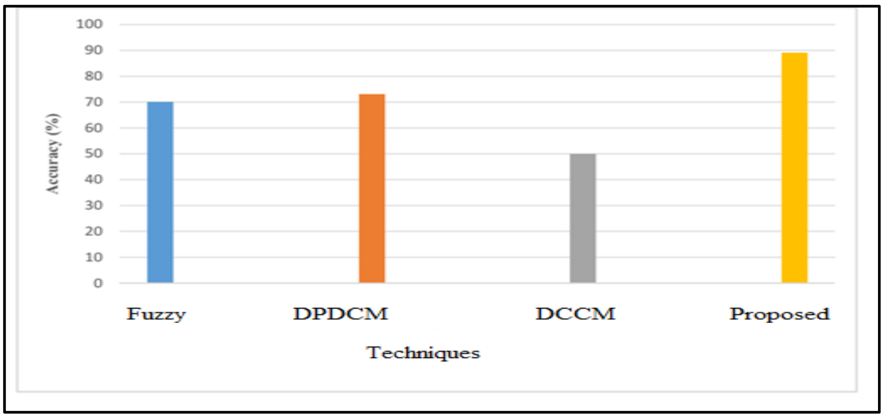

Fig 5: State-of-art result of accuracy

Fig 6 plots the state-of-art result of specificity. The state-of-art methods like Fuzzy, DPDCM, DCCM, and the proposed method evaluate the performance of specificity, which is measured in percentage. The state-of-art techniques like Fuzzy, DPDCM, DCCM, and the proposed method provided 60%, 68%, 72%, and 90% specificity results. From this investigation, the proposed method offers superior classification specificity than other existing methods.

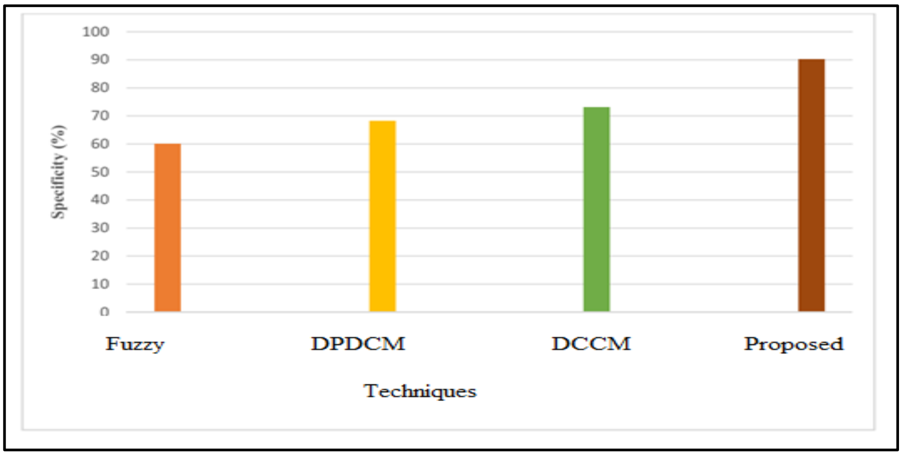

Fig 6: State-of-art result of specificity

The state-of-the-art sensitivity result is displayed in Fig 7. The suggested technique uses state-of-the-art methodologies such as Fuzzy, DPDCM, and DCCM to evaluate the performance of sensitivity, which is evaluated in %. State-of-the-art approaches such as Fuzzy, DPDCM, DCCM, and the suggested method produced sensitivity results of 70%, 60%, 80%, and 91 percent, respectively. Based on the findings, the suggested method outperforms other current methods in terms of sensitivity.

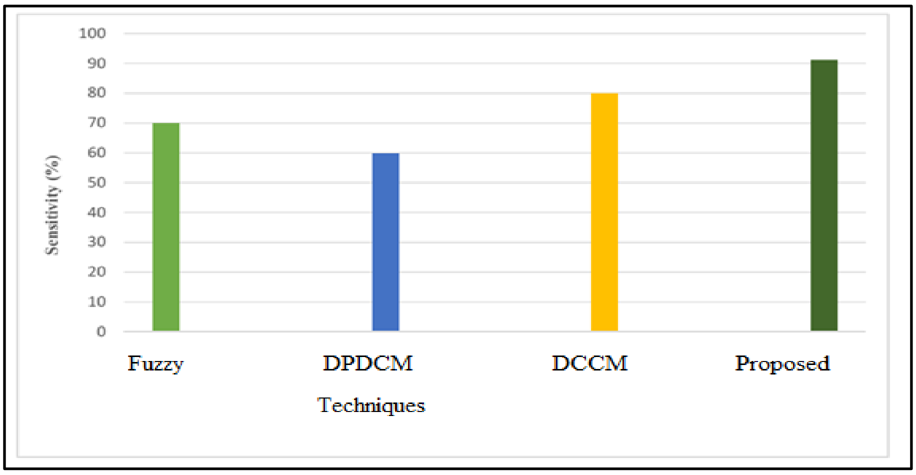

Fig 7: State-of-art result of sensitivity

The state-of-the-art F-measure result is displayed in Fig 8. The suggested technique uses state-of-the-art methodologies such as Fuzzy, DPDCM, and DCCM to evaluate the performance of F-measure, which is evaluated in %. State-of-the-art approaches such as Fuzzy, DPDCM, DCCM, and the suggested method produced F-measure results of 80%, 70%, 82%, and 92%, respectively. Based on the findings, the suggested method outperforms other current methods in terms of F-measure.

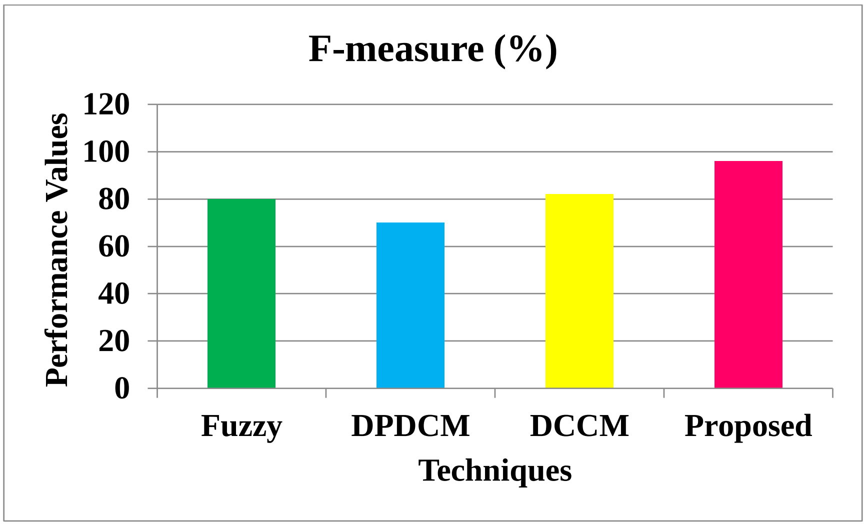

Fig 8: State-of-art result of F-measure

The performance of execution time is displayed in Fig 9. The suggested technique uses state-of-the-art methodologies such as Fuzzy, DPDCM, and DCCM to evaluate the performance of execution time, which is evaluated in %. State-of-the-art approaches such as Fuzzy, DPDCM, DCCM, and the suggested method produced execution time results of 78%, 82%, 89%, and 97%, respectively. Based on the findings, the suggested method outperforms other current methods in terms of execution time.

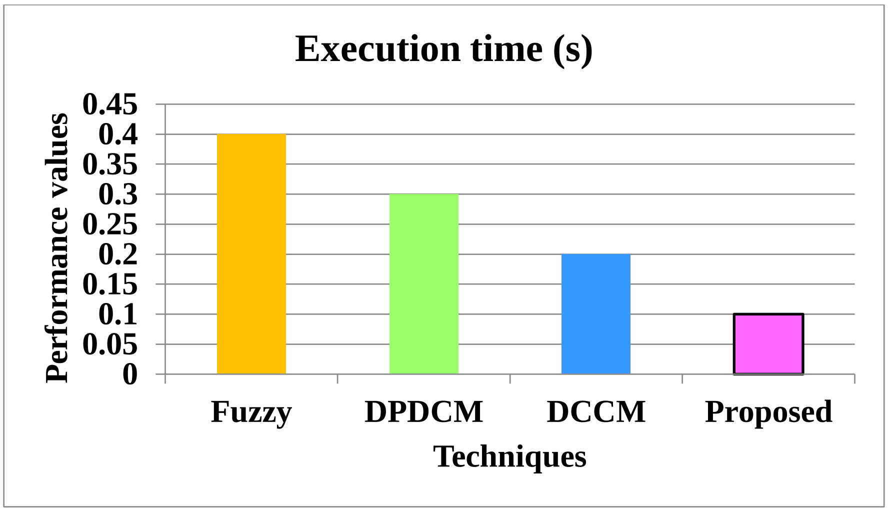

Fig 9: Performance of execution time

The performance of load balancing is displayed in Fig 10. The suggested technique uses state-of-the-art methodologies such as Fuzzy, DPDCM, and DCCM to evaluate the performance of latency, which is evaluated in %. State-of-the-art approaches such as Fuzzy, DPDCM, DCCM, and the suggested method produced load balancing results of 72%, 84%, 79%, and 99%, respectively. Based on the findings, the suggested method outperforms other current methods in terms of load balancing.

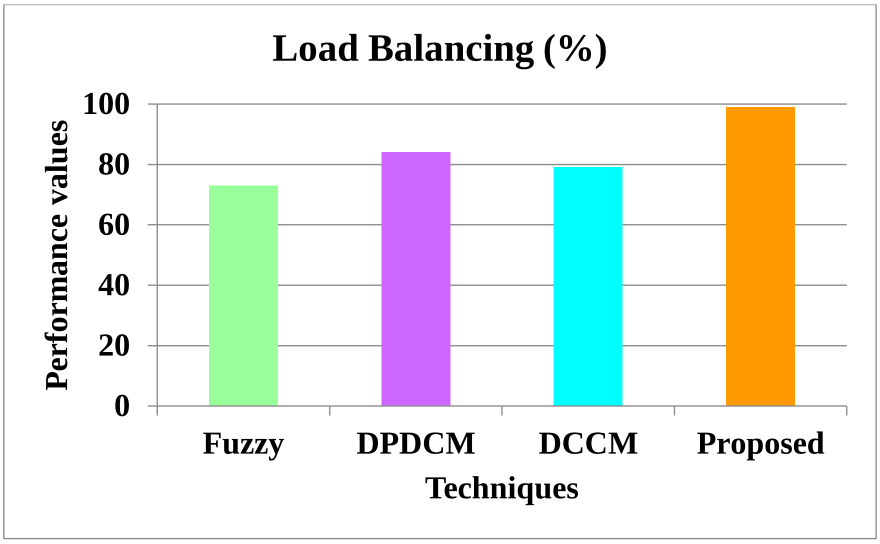

Fig 10: Performance of load balancing

The performance of latency is displayed in Fig 11. The suggested technique uses state-of-the-art methodologies such as Fuzzy, DPDCM, and DCCM to evaluate the performance of latency, which is evaluated in %. State-of-the-art approaches such as Fuzzy, DPDCM, DCCM, and the suggested method produced latency results of 67%, 76%, 84%, and 98%, respectively. Based on the findings, the suggested method outperforms other current methods in terms of latency.

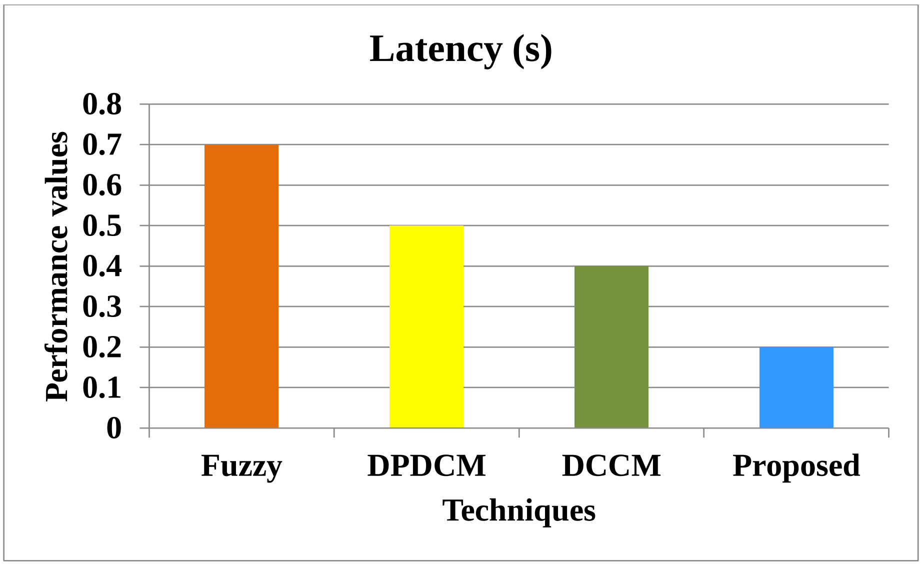

Fig 11: Performance of latency

5. Conclusion:

This research offered a novel machine learning-based strategy that makes use of an SVM classifier. The SVM classifier can be used to classify and analyze data that has been extracted during the data processing step. Data extraction and data scaling are two processes in the data processing process. The data extraction is done using the PCDP-LDA (Principal Component Discriminant Power-based Linear Discriminant Analysis) technique. Meanwhile, the Naive Bayes technique is used to scale the data and partition the collected features into blocks. An experimental analysis is carried out and compared to existing methods. Compared to existing methodologies, our proposed approach delivers excellent categorization and data analytic accuracy. In distributed big data analysis, our technique also delivers superior latency and data load balancing. While compared to the existing methods like Fuzzy, EL, DPDCM, and RF-ELM, the proposed method offers superior accuracy, specificity, sensitivity, and F-measure results.

Compliance with Ethical Standards

Conflict of interest

The authors declare that they have no conflict of interest.

Human and Animal Rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed Consent

Informed consent does not apply as this was a retrospective review with no identifying patient information.

Conflicts of interest Statement: Not applicable

Consent to participate: Not applicable

Consent for publication: Not applicable

Availability of data and material:

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Code availability: Not applicable

Acknowledgement: The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work, under the Research Groups Funding program grant code (NU/RG/SERC/12/13).

References

[1] Ghazal, T.M., Hasan, M.K., Alshurideh, M.T., Alzoubi, H.M., Ahmad, M., Akbar, S.S., Al Kurdi, B. and Akour, I.A., 2021. IoT for smart cities: Machine learning approaches in smart healthcare—A review. *Future Internet*, *13*(8), pp.218.

[2] Thoutam, V., 2021. Physical Design, Origins And Applications Of Iot. *Journal of Multidisciplinary Cases (JMC) ISSN 2799-0990*, *1*(01), pp.26-33.

[3] Farhoumandi, Matin, Quan Zhou, and Mohammad Shahidehpour., 2021. A review of machine learning applications in IoT-integrated modern power systems. *The Electricity Journal*, 34, no. 1, pp. 106879.

[4] Sagheer, A., Mohammed, M., Riad, K. and Alhajhoj, M., 2021. A cloud-based IoT platform for precision control of soilless greenhouse cultivation. *Sensors*, *21*(1), pp.223.

[5] Lakhan, A., Ahmad, M., Bilal, M., Jolfaei, A. and Mehmood, R.M., 2021. Mobility aware blockchain enabled offloading and scheduling in vehicular fog cloud computing. *IEEE Transactions on Intelligent Transportation Systems*.

[6] Qinxia, H., Nazir, S., Li, M., Ullah, H., Lianlian, W. and Ahmad, S., 2021. AI-Enabled Sensing and Decision-Making for IoT Systems. *Complexity*.

[7] Sharma, P.K., Dennison, M. and Raglin, A., 2021. Iot solutions with multi-sensor fusion and signal-image encoding for secure data transfer and decision making. *arXiv preprint arXiv:2106.01497*.

[8] Sadeeq, M.M., Abdulkareem, N.M., Zeebaree, S.R., Ahmed, D.M., Sami, A.S. and Zebari, R.R., 2021. IoT and Cloud computing issues, challenges and opportunities: A review. *Qubahan Academic Journal*, *1*(2), pp.1-7.

[9] Li, P., Chen, Z., Yang, L.T., Zhang, Q. and Deen, M.J., 2017. Deep convolutional computation model for feature learning on big data in internet of things. *IEEE Transactions on Industrial Informatics*, 14(2), pp.790-798.

[10] Li, H., Ota, K. and Dong, M., 2018. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE network*, 32(1), pp.96-101.

[11] Hossain, B., Morooka, T., Okuno, M., Nii, M., Yoshiya, S. and Kobashi, S., 2019. Surgical outcome prediction in total knee arthroplasty using machine learning. *Intelligent automation and soft computing*, 25(1), pp.105-115.

[12] Zhang, Q., Yang, L.T., Chen, Z., Li, P. and Deen, M.J., 2017. Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning. *IEEE Internet of Things Journal*, 5(4), pp.2896-2903.

[13] Jeong, Y.S. and Park, J.H., 2019. Advanced big data analysis, artificial intelligence & communication systems. *Journal of Information Processing Systems*, 15(1), pp.1-6.

[14] Sankaranarayanan, S., Rodrigues, J.J., Sugumaran, V., and Kozlov, S., 2020. Data flow and distributed deep neural network based low latency IoT-edge computation model for big data environment. *Engineering Applications of Artificial Intelligence*, *94*, pp.103785.

[15] Xu, W., Fang, W., Ding, Y., Zou, M. and Xiong, N., 2021. Accelerating federated learning for iot in big data analytics with pruning, quantization and selective updating. *IEEE Access*, *9*, pp.38457-38466.

[16] Hajjaji, Y., Boulila, W., Farah, I.R., Romdhani, I. and Hussain, A., 2021. Big data and IoT-based applications in smart environments: A systematic review. *Computer Science Review*, *39*, pp.100318.

[17] Lee, G.H., Han, J. and Choi, J.K., 2021. MPdist-based missing data imputation for supporting big data analyses in IoT-based applications. *Future Generation Computer Systems*, *125*, pp.421-432.

[18] de Almeida, V.E., de Sousa Fernandes, D.D., Diniz, P.H.G.D., de Araújo Gomes, A., Véras, G., Galvão, R.K.H. and Araujo, M.C.U., 2021. Scores selection via Fisher’s discriminant power in PCA-LDA to improve the classification of food data. *Food Chemistry*, pp.130296.

[19] Mansour, N.A., Saleh, A.I., Badawy, M. and Ali, H.A., 2021. Accurate detection of Covid-19 patients based on Feature Correlated Naïve Bayes (FCNB) classification strategy. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-33.

[20] Smys, S., and Haoxiang, W., 2021. Naïve Bayes and entropy based analysis and classification of humans and chat bots. *Journal of ISMAC*, 3(01), pp.40-49.

[21] Soumaya, Z., Taoufiq, B.D., Benayad, N., Yunus, K. and Abdelkrim, A., 2021. The detection of Parkinson disease using the genetic algorithm and SVM classifier. *Applied Acoustics*, *171*, pp.107528.

[22] Wang, H., Shao, Y., and Xiu, N., 2021. Proximal operator and optimality conditions for ramp loss SVM. *Optimization Letters*, pp.1-16.

[23] Hedrih, K.R.S., 2021. Linear and non-linear transformation of coordinates and angular velocity and intensity change of basic vectors of tangent space of a position vector of a material system kinetic point. *The European Physical Journal Special Topics*, *230*(18-20), pp.3673-3694.

[24] Widodo, A., and Yang, B.S., 2007. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6), pp.2560-2574.

[25] Wang, H., Xu, Z. and Pedrycz, W., 2017. An overview on the roles of fuzzy set techniques in big data processing: Trends, challenges and opportunities. *Knowledge-Based Systems*, 118, pp.15-30.

[26] Li, Y., Gai, K., Qiu, L., Qiu, M. and Zhao, H., 2017. Intelligent cryptography approach for secure distributed big data storage in cloud computing. *Information Sciences*, 387, pp.103-115.

[27] Dolev, S., Florissi, P., Gudes, E., Sharma, S. and Singer, I., 2017. A survey on geographically distributed big-data processing using MapReduce. *IEEE Transactions on Big Data*, 5(1), pp.60-80.

[28] Nguyen, K., Fang, L., Navasca, C., Xu, G., Demsky, B. and Lu, S., 2018. Skyway: Connecting managed heaps in distributed big data systems. *ACM SIGPLAN Notices*, 53(2), pp.56-69.

[29] Anavangot, V., Menon, V.G. and Nayyar, A., 2018, November. Distributed big data analytics in the Internet of signals. In 2018 International Conference on System Modeling & Advancement in Research Trends (SMART). *IEEE*. pp. 73-77

[30] Corizzo, R., Ceci, M. and Japkowicz, N., 2019. Anomaly detection and repair for accurate predictions in geo-distributed big data. *Big Data Research*, 16, pp.18-35.

[31] Li, P., Guo, S., Miyazaki, T., Liao, X., Jin, H., Zomaya, A.Y. and Wang, K., 2016. Traffic-aware geo-distributed big data analytics with predictable job completion time. IEEE Transactions on Parallel and Distributed Systems, 28(6), pp.1785-1796.

[32] Al Najada, H., Mahgoub, I. and Mohammed, I., 2018, November. Cyber intrusion prediction and taxonomy system using deep learning and distributed big data processing. In 2018 IEEE symposium series on computational intelligence (SSCI). *IEEE*.pp. 631-638

[33] Hu, H., Wen, Y., Chua, T.S. and Li, X., 2014. Toward scalable systems for big data analytics: A technology tutorial. *IEEE access*, 2, pp.652-687.

[34] Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R. and Muharemagic, E., 2015. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1), pp.1-21.

[35] Ahmed, E., Yaqoob, I., Hashem, I.A.T., Khan, I., Ahmed, A.I.A., Imran, M. and Vasilakos, A.V., 2017. The role of big data analytics in Internet of Things. *Computer Networks*, 129, pp.459-471.

[36] Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I.A.T., Siddiqa, A. and Yaqoob, I., 2017. Big IoT data analytics: architecture, opportunities, and open research challenges. *ieee access*, 5, pp.5247-5261.

[37] Aryal, A., Liao, Y., Nattuthurai, P. and Li, B., 2018. The emerging big data analytics and IoT in supply chain management: a systematic review. Supply Chain Management: An International Journal.