Vital Annex: International Journal of Novel Research in Advanced Sciences
ISSN: 2751-756X
Volume 04 Number 09 (2025)
https://journals.innoscie.com/index.php/ijnras

Article

A Cloud-Based Enhanced Discretized Support Vector Classifier for Scalable Big Data Prediction

Alaa Abdullhussain Hussain

1. Sumer University, college of Management and Economics, Iraq
* Correspondence: -

Abstract: Big data is a huge amount of data that is such a large amount that it is difficult to process using conventional methods of database and software. When using big data-related applications technical barriers are encountered when moving data between different locations that is costly and requires massive main memory for processing. Big data is a term used to describe interactions and transactions of data in relation to their magnitude and complexity that go beyond the normal technical capability of the capture, organization and processing of data within the cloud. It features real-time processing of data which runs in high-performance clusters. Applications that use big data are designed to share structured and unstructured information. They collect the data in a way that allows for speedier response and reduce the time for classification. Similarly, in this paper, a Discretized Support Vector Classification and Prediction (EEDSV-CP) model is suggested to provide effectual computation upon huge data apps and sharing in a cloud computing environment. Originally, pre-processing was carried out in the EEDSV-CP model using interval equivalence discretization, which aids in the removal of noise and erratic data obtained out of various sources. The computation temporal and spatial complexity are mitigated out by denoising and inconsistizing the data. Furthermore, the EEDSV-CP model employs a supportive vector prediction classifier to categorize data centered upon user query request by employing parallel hyperplanes, with the aim of increasing classification accuracy of customer data requesting on big data. The proposed EEDSV-CP precisely predicts the customer data requesting on big data with the classified data.

Keywords: EEDSV-CP, High-performance clusters, classification, hyperplanes.

Citation: Hussain, A. A. A Cloud-Based Enhanced Discretized Support Vector Classifier for Scalable Big Data Prediction. Vital Annex: International Journal of Novel Research in Advanced Sciences 2025, 4(9), 378-388

Received: 10th Jul 2025
Revised: 16th Aug 2025
Accepted: 24th Sep 2025
Published: 01th Oct 2025

1. Introduction

((Big data is a real-time processing that is data intensive and is run using high performing clusters. Big data communications are used to spread data to various places. This is a very costly process and requires a huge storage space to store the data to run. Big data includes interactions and transaction datasets that are based on the dimensions and complexity that are beyond standard technical capabilities in organizing, capturing and operationalizing data for a rational price in a cloud-based [1].[2] Big data computation and data sharing is accomplished by pre-processing data in the cloud. For big data applications the process of data gathering has increased exponentially.

Big data apps are the gathering and sharing of data that requires more memory consumption. The major challenge with the big data application is the ability to analyse massive amounts of data to extract valuable information or data to help in future tasks. The redundant noises, taken out of differing sources existing in the data are cut out with the aid of pre-opertaionalizion, that mitigates the duration taken for computation and enhances data-sharing. The disseminated data mining on great deal of data-cloud requires least computational overhead and communication costs. Huge data is accounted for by its quantity exceeding the usual range of databases. The huge data concept explains volume, velocity and variety model.[3] Volume is related to the huge size of data that requires to

be handled to extract better data. Velocity feature analyses the big data that is essential to provide learned response within a logical time limit. In the same way, variety refers to the different type of data that compose the quantity of data. The classification of data is widely used for a large amount of effective and efficient means of assigning knowledge and information to users. Though the appearance of large datasets, the traditional classification methods fail to generate required findings. In classifying huge data, the obstacle is made of assessing and understanding exceptionality of huge datasets by retrieving valued geometric and statistical patterns. Because of the broad data related to data processing and availability of benefits, Big Data has achieved great research worth. Large data apps are processed with scalable character of data through Mapduce programming models.

A MapReduce operation is designed to work in single cluster environments that are not used for large-scale disseminated data operationalization through diverse clusters [4]. The MapReduce programming model is made up to handle large amounts of data in parallel by breaking the job down into a series of independent tasks. The job denotes the entire MapReduce program, which is the application of a mapper or reducer to a set of data. Similarly, a task is the application of a mapper or reducer to a segment of data. As a result, the MapReduce job divides the input dataset into independent blocks, which are developed in complete parallel mode by the map tasks.

A single master node runs job tracker instances in the Hadoop MapReduce structure. The master node acknowledges client node job requests, and there are several slave nodes, each running a job tracker instance [5]. The responsibility of a job tracker is to allocate the software configuration to the slave nodes, scheduling the job's module tasks on the task trackers, monitoring them and reallocating tasks to the task trackers when the task fails. The job tracker is also responsible for maintaining the status and for determining the required information for the client while the task trackers perform the tasks as directed by the job tracker. The task tracker creates a task using a Java process so that multiple task cases can run equivalently. Figure 1 depicts the HDFS architecture.

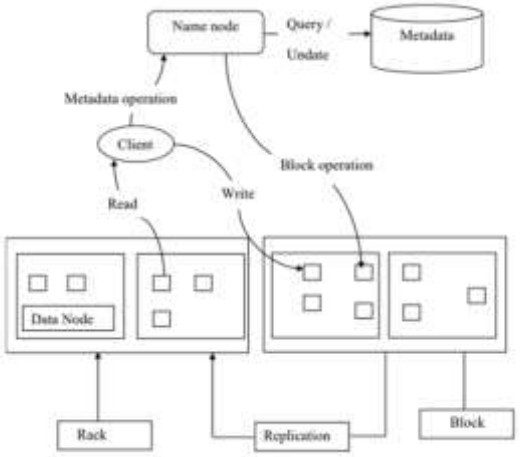

Figure 1. HDFS Architecture.

The Gfarm file is a global distributed model that uses the MapReduce. It disregards the use of block-based storage. Partitioning files into blocks significantly rises the number of metadata, which has an inherent impact on sacred inactivity and bandwidth in broad area networks. The Gfarm file system applies file that depends on storage semantics for increasing the overall performance as it is integrated with Hadoop framework.

2. Materials and Methods

The multiple shared High-End Computing (HEC) clusters are taken as a design environment for G-Hadoop framework. These clusters generally include specialized hardware that are interrelated to maximal performance networkings like InfiniBand. The

main objective of G-Hadoop is to schedule MapReduce apps through numerous data centres connected via broad area networks. G-Hadoop makes use of the Gfarm file framework, which is derived from the disseminated file system. The Gfarm file was created to achieve the needs of offering a global virtual file through numerous administrative scopes. The G-Hadoop framework was designed for broad-place operation, and it provides critical location knowledge for allocating data-aware scheduling between clusters.

Distributed Entropy Minimization Discretizer for Big Data Analysis

In data reduction techniques, discretization is a significant task for data mining operation, designed by facilitating and decreasing ongoing valued data in huge datasets. In this technique, some simple discretization techniques are executed. A dissemination development of the entropy minimization discretizer is intended for using Apache spark platform. The dispersed version of the entropy minimization discretizer is designed by [6] centered upon the Minimum Description Length Principle (MDLP). Discretization algorithm verifies multi-interval extraction of points and the use of boundary points that develop the discretization method in terms of both efficiency and error rate. It is used in distributed environments, establishing its ability across large real-world problems.

The complexity of the discretization algorithm is distributed through the cluster. Discretization is generally calculated by two-time consuming processes that are sorted into candidate points and the evaluation of these points. It determines the minimum entropy cut points through attributes based on the MDLP measure. MDLP is defined as the input parameters such as the dataset, the features indexes for discretizing and the maximal amount of points per partition. Boundary point selection explains the function of choosing points that fall in the class borders. Boundary point performs an independent function on every separation for parallelizing the collection process probable so that a subset of tuples is obtained in each thread. The evaluation is explained for every instance and will evaluate if the featuring index is unique of the index of the earlier point.

Discretizing huge datasets is introduced as a sound multi-interval discretizing way that depends on entropy minimization. Interval Equivalence Discretization (IED) is presented with pre-processing task for denoising and getting rid of inconsistent data taken out of diverse sources. Discretization factor is obtained from big data applications based on interval measures. Then IED value is attained with the help of the measured data applications. Finally, the resultant data are obtained from different big data applications with the removal of noise and inconsistent data that reduce the computation time for data-sharing in cloud setting. The sharing of information is done with the development of discretization algorithm that is accepted between two different nodes for communicating the data.

3. Results and Discussion

Authors in [7] have designed data mining concepts with big data. With the growth of networking and data collection, the size and usage of big data are increasing in all science and engineering areas with physical, biological and biomedical sciences. HACE theorem classifies the features of big data revolution. Big data processing model is designed from data mining viewpoint.

Rough set theory addresses the issues in recognizing pattern and mining data and focalizes around the idea of unique objects through the lower and the upper bound. A parallel matrix-based method is designed by [8] from scientific community for large scale data analysis. A MapReduce-based method is recommended for parallelizing approximations that process the entire data set and fail in missing or non-completed data. To overcome this issue, three different parallel matrix-based methods are utilized for processing the large scale and non-completed data. The challenge comes when the large quantity of data is processed to mine the knowledge for the data mining methods are non-adaptive to novel spatial and temporal requirements. Real world data apps frequently provide class dissemination in which the samples of one class are outnumbered by the samples of the other classes. This occurrence "the class imbalance problem" making learning difficult as standard learning methods do not aptly tackle this state. Many

methods are adopted to cope with the imbalanced datasets in big data scenario by haphazard foresting classifier. Oversampling, under sampling and cost-sensitive learning are adopted by [9] to big data by MapReduce to identify the underrepresented class.

Opinion mining mines meaningful opinions from the data of many social multimedia. It identifies the intent of social networking site consumers. It needs an effective method to gather social multimedia data and to mine meaningful information. A MapReduce function is designed by [10] to mine sentiment information from many unstructured socialmedia platforms text data from social networkings by employing parallel HDFS for saving social hypermedia data and MapReduce functions for sentimental analysis. The method works by collecting data and loading processes. It maintains the stable memory and CPU resources via processing data by HDFS system.

Big data applications facilitate faster processing and information sharing by removing the data with the tridiagonal symmetric matrix. The Cross-validated Bayes classifier model can also be employed for evaluating the real value of diagonal searching results for their equivalent query outcomes derived out of every consumer's request. The MapReduce function works to the Bayes classes derived out of the researching data. It gives accurate prediction analytics on large data to facilitate efficient computation and sharing of information.

The main challenges faced by huge data apps is the massive amounts as well as the extraction of costly data or knowledge to take further actions. Data mining distributed on skydata includes minimal calculation costs and communication costs. The distributed environment is sufficient to use large data sets. Existing large datasets, traditional classification methods have not been able to give better outcomes. Classification performance be ineffective in handing out and partaking data with various apps.

The EDSV-CP model is made out for better computation on huge data apps as well as data allocation in a cloud computing environment. Initially, pre-processing is carried out in the EDSV-CP model centered upon IED, which aids in the removal of noise and inconsistent data taken out of numerous sources. The computational duration and space complexity in data-sharing in the cloud setting are seemingly reduced by denoising and making data inconsistency. For improving the user's classification accuracy concerning data request on huge data, the EDSV-CP model employs a supportive vector prediction classifier to efficaciously categorize the data counting upon the requesting of users' query by parallel hyperplanes. Finally, by using classified data, the EDSV-CP model precisely identifies the customer's data requesting on huge data.

EDSV-CP MODEL

Big data applications can be used to share structured and unstructured information. They collect the data in a way that allows for speedier response and reduce time to classify. The purpose of the EDSV-CP model is designed to facilitate huge data computation and datasharing within the cloud computing setting. This constitutes a massive amount of data that is stored in various formats that are processed by databases that are already established. Enhancements in the accuracy of classifications and the accuracy of predictions for requests from users made on large data are the primary motives to develop the EDSV-CP model, based on current data as well as historical data in the cloud. Huge data apps assembles data for capturing, processing and managing how to share or distribute data among numerous resources. Below the structure of the EDSV-CP.

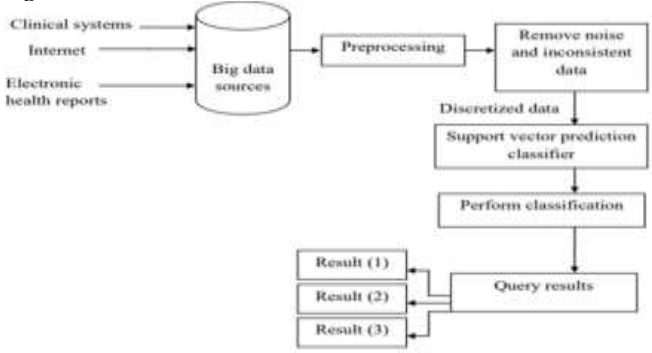

Figure 2

Architecture of EDSV-CP Model.

The suggested EDSV-CP paradigm is responsible for calculating and exchanging information on large amounts of data. The data pre-processing stage in the construction of the EDSV-CP model is the first step, in which the data is discretized with the goal of decreasing the computation time and space complexity of the model. Using the discretized data, it then applies the support vector prediction classifier for the appropriate query findings acquired out of the requests in the next phase.

As in Fig 2, the EDSV-CP initially takes the input out of varying sources, EHR, clinical systems, internet and so on in numerous formats, i.e., flat files, .CSV, ASCII, tables and so on, out of varying sites. Then the massive unstructured data extracted from various sources undergoes pre-processing with the objective of removing the noise and inconsistent data. Therefore, discretized data are extracted from the pre-processing task. Finally, supporting vector prediction classifier model is applied with the discretizing data using the prediction analysis algorithm. This prediction algorithm is employed in Hadoop to calculate and to categorize the data for efficient information sharing in cloud environment.

Hadoop framework uses distributed file system that shares the information with efficient allocation of tasks on different nodes, but it is difficult to provide equal sharing of information. Therefore, the prediction algorithm is implemented in the proposed support vector classification model that produces effective information sharing among different sources. The pre-processing task is applied on the prediction algorithm to extract the structured and unstructured data that provide efficient information sharing in big data applications.

Pre-processing using IED The proposed EDSV-CP model primarily proceeds with pre-processing task. IED is presented with the pre-processing task for denoising and making inconsistent data taken out of numerous sources. Counting on the interval measures, discretization factor is obtained from big data apps and the values under measuring attain IED values. Finally, the resultant data are obtained from different big data applications with removed noise and inconsistent data aiming at reducing the computation time for data-sharing in cloud setting. Therefore, the proposed model initially performs the pre-processing using IED to cut short the computation duration and space complexity. Figure 3 uncovers the block diagram of IED that initially performs pre-processing task to obtain the big data from various sources as they are of different formats.

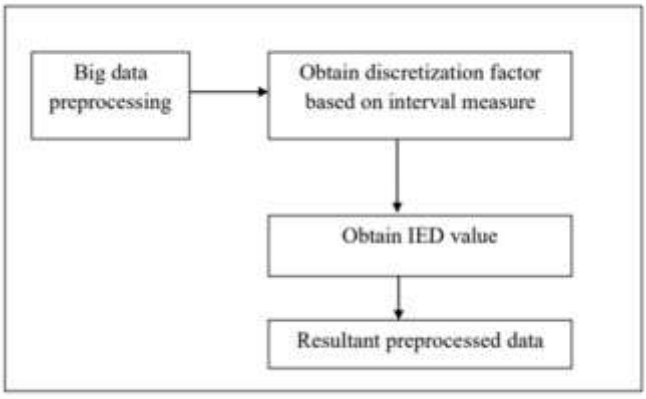

Figure 3: IED Block diagram.

A discretization factor is obtained based on the interval measure. With the resultant value, the IED value is measured to reduce the noise present in the data further. The discretization factor '$\chi 2$' using interval measured in the proposed EDSV-CP model makes an appropriate number of intervals given for the data values. Therefore, changing the continuous data values into the discrete values is expressed as in Equation (1) [11].

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{N} \frac{\left(U_{ij} - EF_{ij}\right)^2}{EF_{ij}} \qquad (1)$$

In Equation (1), the discretization factor is the difference between 'Uij' that corresponds to the number of user requests in 'ith ' interval, 'jth ' class and the expected frequency 'E ij'. The expected frequency is measured as given in Equation (2) [12].

$$EF_{ij} = \frac{(P_i * Q_j)}{N} \quad (2)$$

The expected frequency is the product of number of user requests in the 'ith' interval 'Pi' and the number of user requests in the 'jth' class 'Qj' with respect to the total number of users 'N' in the cloud environment. In Equation (2), 'Qj/N' is the proportion of user requests in the 'jth' class accounting to the total number of users in the cloud environment, and '(Pi∗Qj)/N' corresponds to the number of cloud users in the 'ith' interval. 'χ2' indicates the equivalence degree of the 'jth' class distribution of the adjacent two intervals [13]. The smaller the value of 'χ 2', the more similar the class distribution is said to be. Therefore, the modified discretization factor based on equivalence measure with smaller data interval that contributes to more exact prediction model covering higher prediction rates is expressed as given in Equation (3) [14].

$$M\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{N} \frac{(U_{ij} - EF_{ij})^2}{EF_{ij}} \quad (3)$$

$$M\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{N} \frac{N * \left(\frac{(U_{ij} - P_i * Q_j)}{N}\right)^2}{P_i * Q_j} \quad (4)$$

By using Equation (4), modified discretization factor is measured with the equivalent measure values in big data applications. The pre-processing task using IED contributes towards improving the computation duration and reducing the space complexity in data-sharing in cloud setting.

Data Acquisition Discretized Pre-processing Algorithm (Algorithm 1)

Input: Training Data 'TD = TD1,……TD2, … , TDn', number of user requests 'Uij', expected frequency 'E ij'

Output: Discretized pre-processed data

Step 1: Start

Step 2: Every single Training Data 'TD'

Step 3: Measuring discretization factor using user request and expected frequency

Step 4: Measure modified discretization factor based on equivalence measure

Step 5: End for

Step 6: End

Algorithm 1 describes the steps in discretized pre-processing for evaluating discretized factor values in different big data applications. This evaluation is done for each training data obtained from various big data sources. With the developed pre-processing algorithm, discretization factor and modified factor are evaluated based on equivalence measures. This helps to reduce the noise and make data available in varying sources of data inconsistent. This significantly optimizes the temporal and spatial computation of complexity in information sharing in cloud setting.

Support Vector Prediction Classifier

In EDSV-CP version, guide vector prediction classifier aids to pick out unique sets of huge information every categorized with diverse styles of information. Let an instance of big information from unique resources, namely medical gadget, net, EHR and from different cellular devices be taken into consideration [15].[16] Figure 3 suggests the aid vector prediction classifier hired in EDSV-CP for effective class of large statistics.

Experimental Evaluation of EDSV-CP Model

An Amazon EC2 cloud-based implementation of the EDSV-CP model is used in conjunction with Stanford's Large Network datasets to perform experiments. HDFS's two-layer namespace is used for the tests. HDFS provides a variety of resource settings to accommodate various types of virtual machines. The amount of RAM, CPUs, and local storage available to each kind of virtual machine instance varies.

A total of eight Xeon processors running at 2.33-2.66 GHz, 7 GB of RAM, and 1690 GB of local disc storage are included in the EDSV-CP variant. As a result, in a cloud

computing with HDFS dual-layered namespace, information exchange with massive data is accomplished efficiently. EDSV-CP paradigm with effectual huge data processing and exchange of information is established, providing scalable cloud computing capability. The HDFS dual-layered namespace is used to reduce the computational complexity and expense of the application. As soon as a cloud computing service recognises a user's request for data-sharing, it takes the best data-based judgments and distributes that information to other users without redundancy. With HDFS two-layer namespace, cloud computing efficiently shares information with massive data. Stanford's Large Network dataset collection, which makes considerable use of Amazon's product co-purchasing network, is used to assess DSV-performance. Table 1 lists some of the features utilised in Amazon's product co-purchasing networking.

Table 1. Features utilised in Amazon's product co-purchasing network.

Attributes	Description
N	No. of nodes in social network
n_s	No. of recommendation senders
^{n}r	No. of recommendation recipients
r	No. of recommendations
e	No. of social networking edges
V	No. of product reviews
T	Rate of average product

Evaluating performance of the EDSV-CP and current methods, particularly DM-BD and Flex-Analytics are seemingly linked with Amazon EC2 dataset gatherings, to adequately offer effectiveness to huge data computation. It's made comparisons to DM-BD and Flex-Analytics, which are already in use. User requests for various sized datasets are taken into consideration when CloudSim is used to measure experiment parameters.

Performance Analysis of EDSV-CP Model

The performance of EDSV-CP is in comparison against the current DM-BD and Flex-Analytics methods. In an effort to test the EDSV-CP, the following is used to measure it. i) Computation time ii) Classification accuracy iii) Prediction rate iv) Space complexity

$$CT = \sum_{i=1}^{n} \text{Time}\,(TD_i) \qquad (5)$$

From Equation (5), the computation time is measured on the basis of the time taken to classify the training data 'TDi'. Lower computation time exhibits much improvement in the proposed EDSV-CP model. Table 2 shows the results for computation time in regard to amount of big data being at ranging between 5 and 50 cloud paradigms and it is measured in milliseconds (ms). The proposed EDSV-CP framework, with different amounts of big data instances is taken for experimental purpose using Java language.

Table 2. Results of the Computation Time.

Computation time in (ms)			
Number of Big Data (n)	**Existing DM-BD**	**Existing Flex-Analytics**	**Proposed EDSV-CP**
5	15.34	13.14	10.23
10	17.64	15.68	11.68
15	19.83	16.89	12.14
20	20.84	18.37	14.98
25	21.83	19.42	15.60
30	24.17	21.38	16.84

35	25.64	22.72	17.38
40	26.01	23.84	18.32
45	27.13	24.12	19.67
50	28.91	26.24	21.06

The performance of the EDSV-CP is in comparison to the DM-BD and Flex-Analytics methods. Of Table 3, it is observed that the computation time by EDSV-CP model is reduced compared to that of the existing methods.

Figure 4 reveals the impact of computation time with respect to varying number of instances in the range of 5 to 50. From the Figure 4, it is clear that the proposed EDSV-CP achieves relatively good work in comparison to the DM-BD and Flex-Analytics.

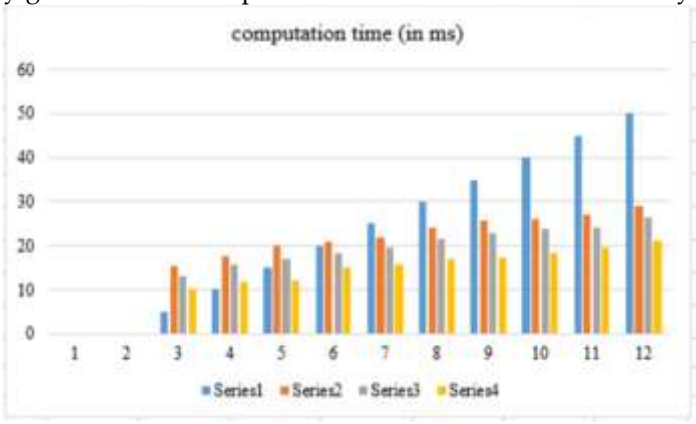

Figure 4. Computation time comparison.

Besides, on increasing the number of huge data, the computation time also gets up, but comparatively, the computation time by EDSV-CP model is significantly reduced than the other methods. This is due to the pre-processing of big data obtained from various sources using IED in EDSV-CP model. Further, based on interval equivalence, the misclassification errors, where interval equivalence function is measured, correspondingly reduces the misclassification errors resulting in efficient split of classes. This, in turn, reduces the computation time during information sharing using EDSV-CP model by 30.93% and 22.00% than DM-BD and Flex-Analytics methods respectively.

The number of correct classifications divided by the total number of big data instances classified in the training dataset represents the Performance Analysis of Classification Accuracy. The classification accuracy 'Ai' of an individual instance in training dataset I is proportional to the number of data correctly classified, as shown in Equation (6).

$$A_i = \frac{DCC}{n} * 100 \qquad (6)$$

where 'DCC' represents the amount of Data Correctly Classified and 'n' represents the total number of data is for evaluation in order to determine classification accuracy. The method is said to be more efficient when the classification accuracy is greater.

Table 3. Results of Classification Accuracy.

Classification Accuracy (%)			
Size of Big-Data (GB)	**Available DM-BD**	**Existing Flex-Analytics**	**EDSV-CP**
200	61.28	66.54	74.3
400	62.87	67.85	76.12
600	64.14	69.67	77.89
800	65.32	71.71	79.12
1000	67.66	72.32	82.34
1200	68.94	74.85	83.78

1400	70.17	76.24	85.67
1600	72.39	78.64	86.12
1800	74.63	79.37	88.47
2000	75.67	81.21	89.36

Table 3 stands for the classification accuracy of EDSV-CP. For determining EDSV-CP performance, there was a comparison made among the classification accuracy of the model with that of the two DM-BD and Flex-Analytics methods. Of Table 3, the classification accuracy of the EDSV-CP is greater in comparison to the existing methods. For tentative assessment, the size of big data is considered in the range of 200 GB to 2000 GB. The impact of classification accuracy employing the three various ways in Fig5. As in Fig5, the EDSV-CP model offers optimized performance in comparison to DM-BD and Flex-Analytics methods.

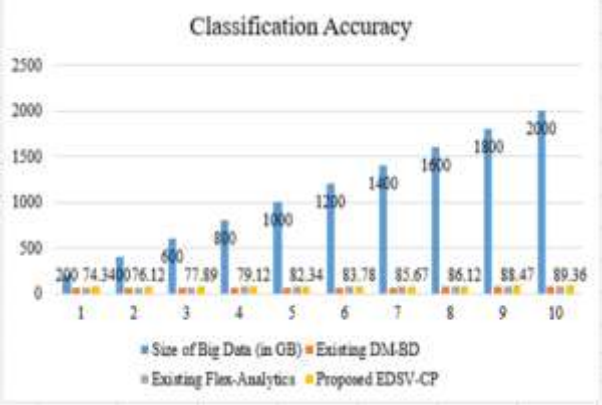

Figure 5. Classification Accuracy.

Further, as the size of big data increases, the classification accuracy is also enhanced, but the classification precision by EDSV-CP is superior. It is ascribed to supportive vector prediction classifier in EDSV-CP cutting short of the misclassification errs extensively.[17] As the presence of dual parallel hyperplanes used, slack variables in the EDSV-CP are added similarly to the misclassification errors are reduced. Therefore, EDSV-CP model classification precision is improved by 20.58% and 11.51% in comparison to DM-BD and Flex-Analytics methods respectively.

Performance Analysis of Prediction Rate

The EDSV-CP anticipates customers' requesting counting on the most recent data and previous experience. In order to forecast future events and trends, the method of collecting information from present user inquiries is known as prediction rate given by Eq. (7).

$$\text{Prediction rate} = \frac{\text{Current data (size)} + \text{Historical facts (size)}}{\text{Size of Big Data}} * 100 \qquad (7)$$

The prediction rate is calculated by using the equation (7) by adding the shape of historical facts in relation to the available data size and the size of large data. The method is called more efficient when the prediction rate is high. The prediction rate when making use of triple varying methods, namely DSVCP models, is expanded in Table 4, DM-BD and Flex Analytics Table 4. Large data sizes from 200 GB to 2000 GB are taken into account for empirical purposes by JAVA code. Of Table 4, it is clear that the prediction rate when making use of the EDSV-CP is greater than the DM-BD and Flex analytics methods.

Table 4. Prediction Rate.

Prediction Rate (in %)			
Size of Big-Data (in GB)	**Existing DM-BD**	**Existing Flex-Analytics**	**EDSV-CP**
200	55.64	59.62	68.72
400	56.48	61.23	69.87

600	57.68	63.45	71.24
800	59.12	65.82	73.48
1000	60.58	67.15	75.68
1200	62.31	68.98	77.95
1400	65.48	70.23	79.82
1600	65.89	72.34	81.34
1800	67.42	74.97	82.38
2000	69.83	75.81	84.33

Fig6 unveils the prediction percentage vs varying volumes in the range of 200 GB to 2000 GB. As in the Fig6, the EDSV-CP employing the prediction percentage gives optimal performance than that of the other dual methods.

Figure 6. Prediction Accuracy.

This support vector is also due to the prediction classification, including parallel hyperplanes to limit hyperplan that passes through the original by selecting the optimal hyperplane to classify large data. Therefore, the prediction rate when using EDSV CP is better than 23.33% and 12.62% than the Flex analytics method.

Performance Analysis of Space Complexity

In EDSV-CP model, the space complexity is a measure of the amount of working storage that the discretized pre-processing algorithm needs. In other words, it is the memory that is required to run the algorithm. The less the memory space is, the more effective the space complexity in the proposed method

4. Conclusion

EDSV-CP model is made up for achieving better huge data processing and sharing data in cloud computing platform in this section. Improving the precision of classification and the prediction rate of the customer data requesting in cloud platform is the core goal. The EDSV-CP starts with performing data pre-processing to efficiently denoising and making datum in datasets inconsistent, which actually minimizes computation time and space complexity. After data pre-processing is performed, the EDSV-CP employs a supportive vector prediction classifier in order to correctly classify big data in the cloud. By providing improved researching and prediction precision of customer's data requesting on big data, the EDSV-CP significantly reduces misclassification errors. The efficiency of the EDSV-CP model is confirmed based on the parameters like computation time, space complexity, classification precision and rates of prediction. The experimental findings show that the developed EDSV-CP model provides better performance with higher classification accuracy and reduces the space complexity in comparison with state-of-the-art methods. Moreover, LFR-CM is presented for implementing big-data operations in hand-in-hand mode for information sharing in cloud environment.

REFERENCES

[1] V. Casola, A. De Benedictis, J. Modic, M. Rak and U. Villano, "Perservice Security SLA: a New Model for Security Management in Clouds," 2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2016, pp. 83-88.

[2] Centre of Protection of National Infrastructure, Information Security Briefing: Cloud Computing, [Online]. Available: https://www.cpni.gov.uk/system/files/documents/1f/8d/cloud-computing-briefing.pdf

[3] C. Yang, Q. Huang, Z. Li, K. Liu and F. Hu, "Big Data and Cloud Computing: Innovation Opportunities and Challenges," International Journal of Digital Earth, vol. 10, no. 1, pp. 15-53, 2017.

[4] J. Cao, H. Cui, H. Shi and L. Jiao, "Big Data: A Parallel Particle Swarm Optimization-Back-Propagation Neural Network Algorithm Based on MapReduce," PLoS ONE, vol. 11, no. 6, pp. e0157551, 2016.

[5] I. Ha, B. Back and B. Ahn, "MapReduce Functions to Analyze Sentiment Information from Social Big Data," International Journal of Distributed Sensor Networks, vol. 11, no. 6, pp. 1-11, 2015.

[6] D. Agrawal and P. Kulurkar, "A cloud-based system for enhancing security of android devices using modern encryption standard-II algorithm," International Journal of Innovations and Advancement in Computer Science, vol. 5, no. 4, pp. 60-69, 2016.

[7] E. Ezhilarsan and M. Dinakaran, "Secure Big Data Storage Using Training Dataset Filtering-K Nearest Neighbour Classification with Elliptic Curve Cryptography," Journal of Computational and Theoretical Nanoscience, vol. 15, no. 6-7, pp. 2437-2442, 2018.

[8] J. Cao and Z. Lin, "Extreme Learning Machines on High Dimensional and Large Data Applications: A Survey," Mathematical Problems in Engineering, vol. 2015, pp. 1-21, 2015.

[9] J. Chase, D. Niyato, P. Wang, S. Chaisiri and R. Ko, "A Scalable Approach to Joint Cyber Insurance and Security-as-a-Service Provisioning in Cloud Computing," IEEE Transactions on Dependable and Secure Computing, 2017.

[10] P. D. Diamantoulakis, V. M. Kapinas and G. K. Karagiannidis, "Big Data Analytics for Dynamic Energy Management in Smart Grids," Big Data Research, vol. 2, no. 3, pp. 94-101, 2015.

[11] I. D. Dinov et al., "Predictive Big Data Analytics: A Study of Parkinson's Disease Using Large, Complex, Heterogeneous, Incongruent, Multi-Source and Incomplete Observations," PLoS ONE, vol. 11, no. 8, pp. e0157077, 2016.

[12] B. Kalyani and Y. V. Reddy, "Big data and cloud-based health care records monitoring using deep learning technology," Journal of Critical Reviews, vol. 7, no. 12, pp. 5192-5201, 2020.

[13] G. Gao, R. Li, H. He and Z. Xu, "Distributed caching in unstructured peer-to-peer file sharing networks," Computers and Electrical Engineering, vol. 40, no. 2, pp. 688-703, 2014.

[14] S. Garcia, J. Luengo, J. A. Sáez, V. Lopez and F. Herrera, "A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 4, pp. 734-750, 2013.

[15] C. S. Dule and H. A. Girijamma, "Content an Insight to Security Paradigm for Big Data on Cloud: Current Trend and Research," International Journal of Electrical and Computer Engineering, vol. 7, no. 5, pp. 2873-2882, 2017.

[16] M. H. U. Rehman and A. Batool, "The Concept of Pattern based Data Sharing in Big Data Environments," International Journal of Database Theory and Application, vol. 8, no. 4, pp. 11-18, 2015.

[17] S. Ramírez-Gallego et al., "Data discretization: taxonomy and big data challenge," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 6, no. 1, pp. 5–21, 2016, doi: 10.1002/widm.1173.