A Review of Data Mining, Big Data Analytics, and Machine Learning Approaches

Francisco Pedro
University of Minho, Braga, Portugal.
franciscobraga1211@gmail.com

Correspondence should be addressed to Francisco Pedro : franciscobraga1211@gmail.com.

Article Info
Journal of Computing and Natural Science (http://anapub.co.ke/journals/jcns/jcns.html)
Doi: https://doi.org/10.53759/181X/JCNS202303016
Received 10 October 2022; Revised from 12 December 2022; Accepted 02 April 2023.
Available online 05 October 2023.

Abstract – The phenomenon of economic globalization has led to the swift advancement of industries across diverse domains. Consequently, big data technology has garnered increasing interest. The generation of network data is occurring at an unparalleled pace, necessitating the intelligent processing of vast amounts of data. To fully leverage the value inherent in this data, the implementation of machine learning techniques is imperative. The objective of machine learning in a vast data setting is to identify particular rules that are concealed within dynamic, variable, multi-origin heterogeneous data, with the ultimate aim of maximizing the value of the data. The integration of big data technology and machine learning algorithms is imperative in order to identify pertinent correlations within intricate and dynamic datasets. Subsequently, computer-based data mining can be utilized to extract valuable research insights. The present study undertakes an analysis of deep learning in comparison to conventional data mining and machine learning techniques. It conducts a comparative assessment of the strengths and limitations of the traditional methods. Additionally, the study introduces the requirements of enterprises, their systems and data, the IT challenges they face, and the role of Big Data in an extended service infrastructure. This study presents an analysis of the probability and issues associated with the utilization of deep learning, including machine learning and traditional data mining techniques, in the big data analytics context.

Keywords – Machine Learning, Big Data, Data Mining, Big Data Analytics, Traditional Data Mining.

I. INTRODUCTION

Classification techniques, clustering methods, association rule approaches, and regression methods are the four main building blocks of data mining. In this scenario, clustering is used to locate processes that are similar to one another, and regression is performed to see whether the data can be modelled with the least amount of error. The first step in selecting a learning dataset is establishing a classification model, which allows the data to be automatically segmented into many classes. Naïve Bayesian classification techniques, decision trees, artificial neural networks (ANN), and support vector machines (SVM) are only few of the numerous techniques available for use in machine learning classification. **Fig. 1** displays the normal grouping and illustrates a machine learning classification task.

The process of dimensionality reduction is known to aid in the efficient handling of high-dimensional data, enabling tasks such as classification, communication, visualization, and storage. Principal component analysis (PCA) is the predominant technique employed in the dimensionality reduction realm. PCA is a straightforward technique that identifies the principal axes of alteration within a given dataset. It accomplishes this by determining the coordinates of each data point along these axes, thereby providing a representation of the data in terms of these dimensions. The initial principal component is referred to as the direction that exhibits the highest projected variance. According to Tang et al. [1], the second principal component is defined as the orthogonal locus, which captivates the second largest predictable variance, and this process continues for subsequent principal components. PCA is a valuable technique in cases where the dataset contains a significant number of variables and there exists redundancy among them. In the present scenario, redundancy refers to the presence of correlation among certain variables. Due to the presence of redundancy, PCA can effectively diminish the observable variance to a diminished principal component set.

Factor analysis is an alternative technique utilized for the purpose of reducing dimensionality. The comprehension of the fundamental causes for the associations observed among a set of variables is valuable. Factor analysis is primarily utilized for the purpose of diminishing the quantity of variables and identifying patterns within the interrelationships among variables. Hence, factor analysis is frequently employed as a technique for identifying underlying structures or reducing data. Its primary purpose is to identify latent variables that underlie observed variables and minimize the quantity of interrelated variables. The technique posits that the observed data is generated by certain latent variables that are not directly observable. It is postulated that the information under consideration could be defined as a linear amalgamation of

the underlying variables and a certain degree of random variation. Dimensionality reduction can be achieved when the latent variable number is minimal compared to the variable number available within the observed data. In practical scenarios, a common practice is to allocate 75% and 25% of the data for training and validation purposes, respectively. The prevalent approach, particularly in the domain of neural networks, involves partitioning the dataset into three distinct subsets, namely training, validation, and testing.

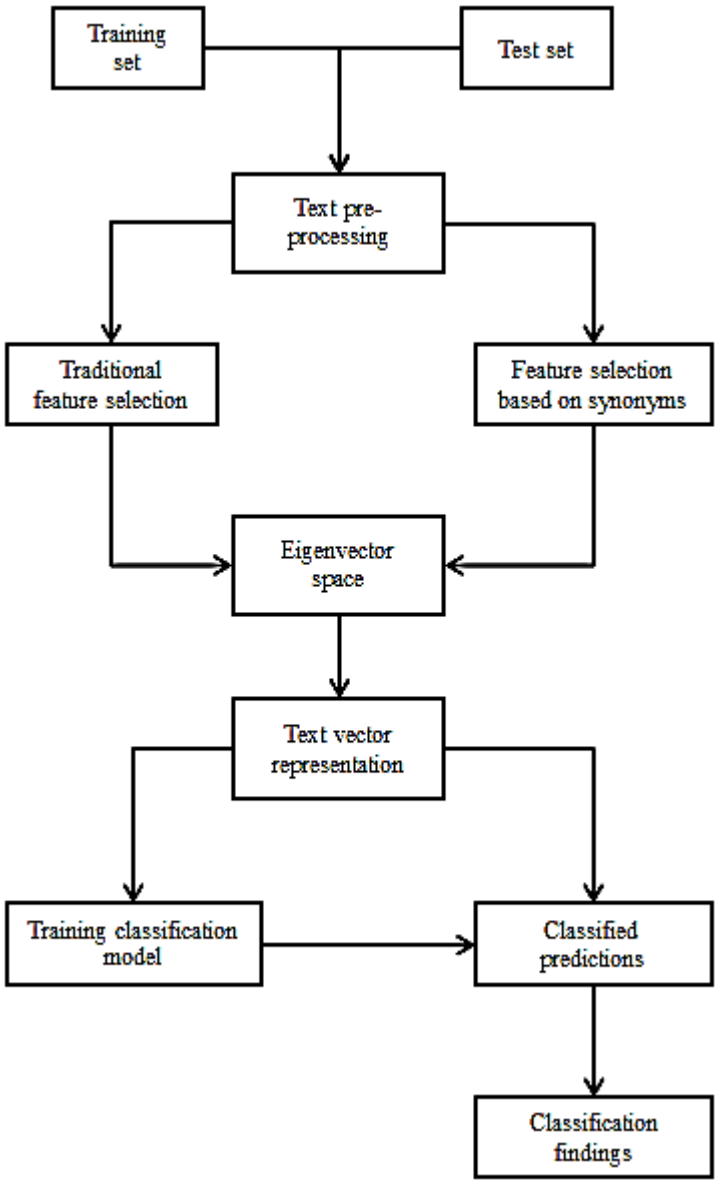

Fig 1. A flow chart depicted the classification task in machine learning.

According [2], the data obtained from testing will not be utilized during the modeling stage. The utilization of k-fold cross-validation is a prevalent approach in estimating the efficacy of a classifier due to its ability to mitigate the over-fitting issue. K-fold cross-validation involves the random partitioning of the original dataset into k distinct and non-overlapping subsets, commonly referred to as "folds". The process of training and testing is repeated k times. According to [3], every sample is utilized an equal number of time used for trainings and only a single time for testing. The process of normalization holds significant value in the context of classification algorithms that employ distance metrics or neural networking system such as nearest-neighbor clustering and classification. Normalization is a technique employed in distance-based methods to mitigate the impact of attributes with large initial ranges, such as income, which may otherwise dominate the influence of attributes with minor initial ranges, e.g., binary attributes. Various techniques exist for normalization of data, such as z-score normalization, min-max normalization, and decimal scaling normalization.

The objectives of these articles have been organized as follows: Section II presents a review of machine learning and data mining methods. In this section, we presents a discussion of k-means, k-prototypes, k-modes, and clustering analysis k-nearest neighbors, SVM, trees and logistic regression, Naïve Bayes, neural networks, deep learning, and comparison of

various techniques and ensemble techniques. Section III focused on machine learning and data mining in big data analytics. This section will present a discussion of representation learning approaches, supervised approaches, unsupervised approaches, and deep architectures approaches. Lastly, Section IV presents a conclusion to the articles.

II. MACHINE LEARNING AND DATA MINING METHODS

Cluster analysis, k-prototypes, k-modes, and k-means

The categorization of clustering techniques encompasses several distinct types, including hierarchical, partitioning, constraint-based, density-based, grid-based, and model-based methods. According to Hajjar, Aldabbagh, and Dimitriou [4], clustering exhibits a primary benefit over classification in its capacity to adjust to modifications and facilitate the identification of significant characteristics that differentiate distinct clusters. An effective clustering technique is one that yields clusters of superior quality, characterized by elevated levels of intra-class resemblance and reduced intra-class resemblance dimensions. The clustering effectiveness is reliant on the methodological suitability utilized within data, the dissimilarity measures employed, and its process of execution. The effectiveness of the clustering approach is also assessed on the basis of its capability to detect a number of latent patterns, or all.

The various forms of information employed in the analysis of clusters incorporate the binary variables, categorical variables, the ordinal variables, mixed variables, and the interval-scaled variables. The k-means algorithm employs a greedy iterative methodology for identifying clusters that minimize the sum of squared errors (SSE). According to Duarte [5], it is conceivable that the convergence of the algorithm may lead to a local optimum as opposed to a global optimum. The k-means algorithm is characterized by several significant properties, as outlined in [6]: 1) The algorithm demonstrates high efficiency in handling voluminous data sets. 2) It is limited to processing solely numerical values. 3) The clusters exhibit convex shapes. It is necessary for users to predefine the k value that signifies the cluster number. The algorithm may converge to a local extremum. The utilization of deterministic annealing and genetic algorithms are viable methods for determining the global optimum.

The k-means is not effective for the evaluation of categorical set of data, whereas k-modes are specifically designed to handle categorical data by utilizing modes. The k-modes algorithm employs novel dissimilarity metrics to handle categorical entities and employs a frequency-based approach to revise the cluster modes. According to Gao, Hu, and Chu [7], the k-prototypes algorithm is capable of handling a combination of both numerical and categorical data.

k-Nearest Neighbors

This algorithm identifies a subset of k items from the learning dataset that exhibit the shortest distance to the test item. The classification decision is then made based on the prevalence of a specific category within this local neighborhood. The k-Nearest Neighbor (k-NN) algorithm entails the classification of an object based on the category of its nearest neighbors. The algorithm determines the k training number, which signifies the nearest proximities to the hidden cases and subsequently selects the classification that appears most frequently among these k cases. There are some pivotal factors that impact the efficacy of k-NN. The selection of k is a determining factor. When the value of k is small, the outcome may exhibit sensitivity towards noise points.

Conversely, in the event that k is excessively large, the vicinity could potentially encompass an excessive number of points originating from alternative categories. Cross-validation can be utilized to derive an approximation of the optimal value for k. According to Sewwandi, Li, and Zhang [8], it can be inferred that higher values of k exhibit greater resistance to noise when an adequate number of samples are available. The algorithm utilized in taxonomy is an uncomplicated instance-oriented learning technique. Notwithstanding its modesty, it has the potential to yield commendable performance on certain problems. According to Tang, Chang, and Li [9], the k-NN algorithm possesses significant properties. 1) The implementation and usage of this system is straightforward. 2) A significant amount of storage space is required to accommodate all of the objects.

Support Vector Machine

The Support Vector Machines (SVM) algorithm is a type of supervised learning technique that is commonly employed for the purpose of classification and regression tasks. Support Vector Machines (SVM) have demonstrated efficacy in addressing computational challenges that are characterized by sparsity, nonlinearity, and high dimensionality. One benefit of this approach is that the construction of the model solely relies on support vectors, as opposed to the entire training dataset. Therefore, the magnitude of the training dataset typically does not present an issue. Moreover, the model exhibits greater robustness to outliers as it solely relies on the support vectors for its construction. One drawback of the algorithm is its susceptibility to the selection of tuning options, such as the specific transformations to be executed. This results in increased time consumption and heightened complexity in utilizing the optimal model. An additional drawback pertains to the fact that the alterations are executed not only during the construction of the model but also when evaluating novel data. This results in a high computational cost. The SVM algorithm is capable of processing both numerical and nominal data types, and it can perform classification tasks on both binary and multiclass target variables, as documented in [10].

The advancement of SVM has occurred within the Statistical Learning Theory (SLT) framework, as evidenced by sources such as Feng and Wu [11]. Initially, we provide a succinct overview of fundamental concepts pertaining to the theory. The field of SLT presents the formulation of the problem of supervised learning in the following manner. The

present study involves a given training set of data x_1, y_1 to x_l, y_l within Rn by R, which has been sampled in accordance with unidentified probability distribution $(x, y)P$. Additionally, $(y, f(x))V$ (the loss function) has been employed to evaluate the error incurred when $f(x)$ is projected in substitute to the real y value, for a given x. The crux of the issue lies in determining a function f that can effectively minimize the anticipated error on novel data, thereby optimizing the expected error: $\int P\,(x, y)(\,y, f(x))V \quad dxdy$. As the value of $P(x, y)$ remains unknown, it is necessary to employ an induction principle to deduce a function that decreases the probable error from the present training datasets. The methodology employed involves the utilization of Empirical Risk Minimization (ERM) across a range of potential functions, which are commonly referred to as the hypothesis space. In a formal manner, this can be expressed as the minimization of the empirical error: $\frac{1}{l}\sum_{i=1}^{1} y_i, f(x_i)V$.

Assuming that the function f is confined to a specific hypothesis space H. A crucial inquiry pertains to the proximity between the solution's empirical error that represents the empirical error minimizer, and the minimum of the probability error, which could be obtained through different functions from H. The theory presents a fundamental outcome which specifies the circumstances where two errors show proximities to each other. In addition, it provides limitations of probability on the available gap between probability and empirical errors, shown in Theorem 1. The aforementioned limits are expressed with respect to a metric of intricacy of the H hypothesis spaces. Specifically, as the complexity of H increases, the probability of the empirical and anticipated errors being further apart also increases, as demonstrated in theorem 1 provided below.

> *Based on the bounds provided by the theory, it happens that it is plausible to enhance ERM inductive rules based on the consideration of the hypothesis space structure ($H_1 \subset H_2 \subset \ldots \subset H_m$), composed of ordered levels of complexity (i.e. H_{i+1} has the highest complexity compared to H_i).*

Enterprise Risk Management (ERM) is executed in all of these domains, and the ultimate selection of a solution could be made by utilizing the above mentioned limitations. The concept of conducting ERM on a nested sequence of hypothesis spaces or structures is commonly referred to as Structural Risk Minimization (SRM), as stated by Liu, Liu, and Chen [12]. A crucial inquiry that emerges in the field of SLT pertains to the quantification of the "complexity" of a hypothesis space. This measure is essential in selecting the ultimate solution to the different learning issues, as we previously deliberated. The article referenced by Deng, Gazagnadou, Hong, Mahdavi, and Lyu [13] delves into the topic of quantifying the hypothesis space's "complexity", and provides recommendations for conducting empirical measurements of said quantities. The present discourse provides a succinct depiction of the "complexity" parameters deliberated upon in [14].

The initial parameter under consideration is a conventional one within the SLT field as introduced by Vapnik and Izmailov [15], and is represented as the VC dimensionality of a particular functional collection. This represents the combinatorial measure, which delineates the function set aptitude to fragments of a wide-variety of points. The available distance between the probability and empirical errors can be bounded by utilizing the VC dimension of a hypothesis space H. Provided the H hypothesis space composed of the V VC-dimensionality, it can be stated that:

> *The minimum expected error (L) achievable with different functions from H and the Lemp (minimal empirical error) are constrained such that the probability of their difference being greater than η is equal to zero.*

The validity of Theorem 1 is not contingent upon distribution of probability of P (x, y) datasets. The literature reviews in [16] posit that the P(x,y) distribution into their analysis, which is a concept that has been previously explored in literature, as evidenced by the work of Vapnik in 1998 and related references. The authors establish an upper bound on the difference between expected and empirical error by utilizing a "complexity" metric that accounts for the distribution P(x,y). The presented bounds exhibit greater precision than those of theorem 1, albeit necessitating familiarity with the complexity measure that is contingent on the distribution. The objective of the study conducted in [17] was to propose a feasible experimental framework for the computation of the complexity measure that is contingent on the distribution, as postulated by their theory. In their work, Bellotti, Matousek, and Stewart [18] provide an overview of the fundamental theoretical underpinnings that have informed the development of learning machines, including Support Vector Machines (SVM). They also propose potential avenues for further research that may enhance the existing theory and, by extension, improve the efficacy of SVM. One such possibility involves leveraging the theory to optimize SVM parameters, such as regularization and kernel parameter C.

Trees and Logistic Regression

The two primary categories of decision tree employed in data mining include: The two trees commonly used in data analysis are classification trees, which are utilized to predict the class to which a given set of data belongs, and regression trees, which are used to predict a real number outcome. The utilization of classification trees and regression trees offers distinct methodologies for the purpose of prediction [19]. In the process of building a tree, various metrics such as the statistical relevance of Gini index, data gains, among others, could be utilized to assess the effectiveness of split. The construction of a decision tree often results in the emergence of numerous branches that are indicative of anomalies in the

training data, which may be attributed to the presence of noise or outliers. The issue of overfitting the data can be mitigated through the implementation of tree pruning techniques. Trees that have undergone pruning tend to exhibit reduced size and complexity, thereby rendering them more easily comprehensible.

According to Da Silva et al. [20], they typically exhibit superior speed and accuracy in the classification of independent test data. There exist two distinct methods for pruning a tree: There are two common techniques for decision tree pruning: post-pruning and pre-pruning. Pre-pruning involves stopping the construction of the tree early and removing branches that do not improve its performance. On the other hand, post-pruning involves removing sub-trees from a completely developed tree that does not improve its accuracy. The adoption of a post-pruning strategy, also known as backward pruning, is a common practice subsequent to the construction of a full tree, as opposed to a pre-pruning approach, also known as forward pruning, as stated by [21]. Conditional inference trees and recursive partitioning trees are nonparametric models that can be applied to both regression and classification tasks. These models are highly straightforward and flexible to understand; however, they are liable to over-fitting.

According to Greenwell [22], conditional inference trees exhibit lower susceptibility to bias compared to recursive partitioning trees. Logistic regression is a statistical model that involves the use of regression analysis to investigate the available connection between a classified dependent variable and a single or multiple independent variables. This approach exhibits computational efficiency, ease of implementation, strong knowledge representation capabilities, and straightforward interpretability. Nonetheless, it is susceptible to under-fitting and might present minimized accuracy levels.

Naïve Bayes

Naïve Bayes represents an algorithm of classification that eschews the use of explicit representations such as decision trees or rules. Instead, it employs probabilistic theorem to present the most feasible classifications. The algorithm is capable of operating effectively with limited data and categorical variables. The Naive Bayes algorithm possesses significant properties, as outlined by Zhang and Ling [23]. There are two notable advantages to this approach: firstly, the construction process is straightforward and training can be accomplished with ease and efficiency; secondly, it boasts a high degree of scalability. The Naive Bayes classifier is renowned for its straightforwardness, expedient computational capabilities, and commendable classification efficacy. In reality, it frequently exhibits superior performance compared to more intricate classifiers, despite the underlying presumption of independent predictors being significantly untrue.

Table 1. Characteristics of neural networks

Prospect of extrapolation concerns	Initially, it is noteworthy that despite the ability of neural networks to generalize from a given set of examples, the prospect of extrapolation remains a significant concern. In the event that a network solely encounters cases falling within a specific range, its prognostications beyond this range may be entirely void of validity.
Lack of inherent mechanisms for variable selection	It should be noted that neural networks lack an inherent mechanism for variable selection. Consequently, it is imperative to exercise prudence in the evaluation of predictors. The amalgamation of regression and classification trees comprised of other dimension reduction approaches such as PCA (principle component analysis), is frequently employed to discern crucial predictors.
High flexibility degree	Thirdly, the neural network's high degree of flexibility is contingent upon the availability of ample training data. Insufficient training set size can lead to poor performance of a neural network, even in cases where the connection between predictors and the response is straightforward.
Faced with technical issues	The fourth concern pertains to a technical issue, namely the potential hazard of acquiring weights, which may result in a local optimum as opposed to the global optimum.
Necessitates more computation and runtime	Neural networks entail a significant amount of computation and necessitate a longer runtime in comparison to alternative classifiers. The computational time experiences a significant increase as the quantity of predictors expands.

This benefit is particularly relevant in scenarios where the quantity of predictors is substantial. There exist additional attributes pertaining to the algorithm. This algorithm necessitates a substantial quantity of data to achieve optimal outcomes. In cases were the probabilistic class is not available from the training dataset, this algorithm assumes a probability of zero for a new record that includes that category as a predictor. This circumstance may pose an issue if the infrequent predictor value holds significance. Ultimately, optimal results are achieved in situations where the objective is to classify or rank data entries based on their likelihood of being associated with a particular category. Nevertheless, in cases where the objective is to accurately assess the likelihood of belonging to a particular class, this approach yields highly prejudiced outcomes. Due to this rationale, the Naive Bayes approach is infrequently employed in the field of credit scoring.

Neural Networks

The ANN refers to a computational framework employed in the classifications and prediction activities. The algorithms utilized in neural networks possess an inherent parallelism. The utilization of parallelization techniques can enhance the efficiency of computational procedures. Furthermore, a number of methodologies have been recently devised for the retrieval of rules from neural networks that have undergone training. This statement pertains to the utilization of neural networks in the realm of data mining for the purposes of classification and prediction, as referenced in [24]. **Table 1** highlights the significant characteristics of neural networks.

Backpropagation is widely recognized as the most commonly used neural network algorithm. The backpropagation algorithm employs a gradient descent technique. The target value in a machine learning context can refer to either the identified class label of training tuples in continuous value or classification problems within prediction tasks, as stated in [25]. To identify the appropriate size of the hidden layer, it is fundamental to strike a balance between underfitting and overfitting. Insufficient utilization of nodes may not be adequate in capturing intricate relationships. Conversely, an excessive number of nodes can lead to the issue of overfitting. According to a commonly used heuristic, the initial number of nodes in a predictive model should be set to "p" (the number of predictors), and subsequently adjusted incrementally while monitoring for signs of overfitting. Neural networks possess several benefits, such as their notable predictive capabilities, capacity to tolerate noisy data, and aptitude for classifying patterns that were not included in their training dataset. Decision trees are a suitable option in situations where the understanding of the associations between attributes and classes is limited. Unlike the majority of decision tree algorithms, these models are highly compatible with inputs and outputs that are continuous-valued.

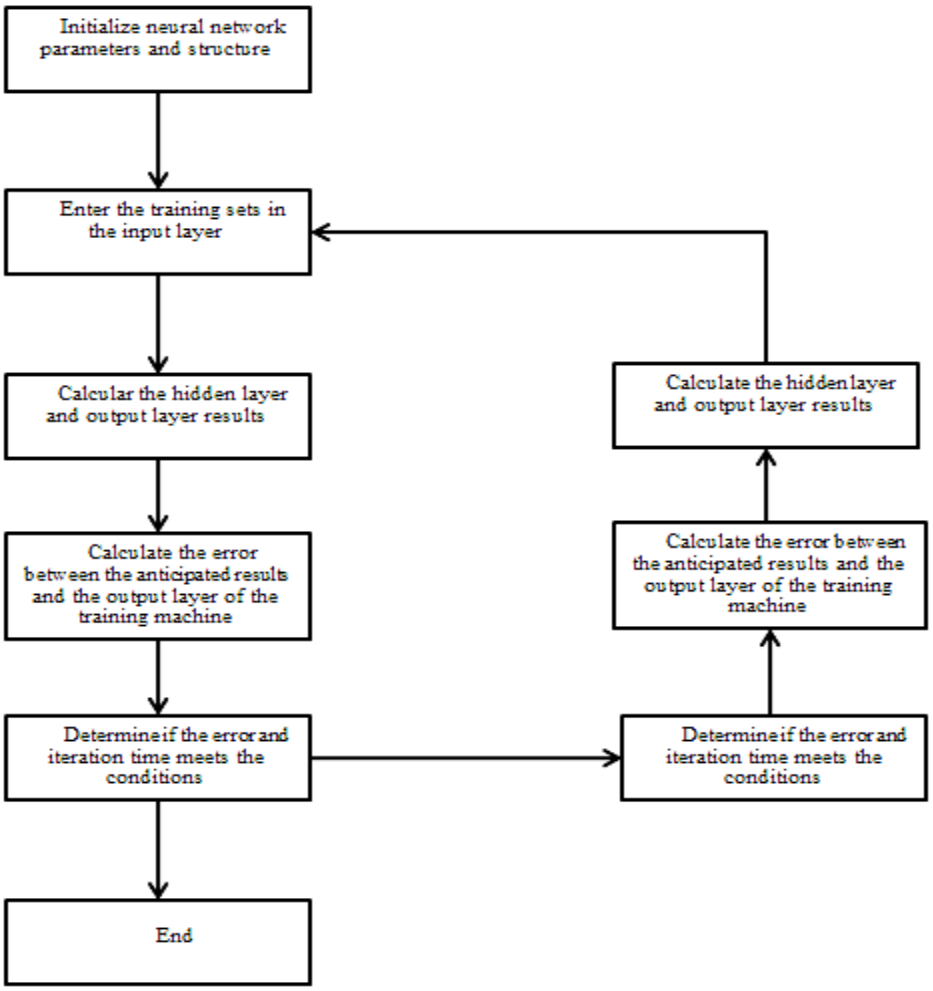

Fig 2. Schematic diagram of the neural network training

The features of the BP Neural Network (BP) are as follows: For the most part, it opts for the decentralized manner of data storage that is well-suited to massively parallel computations. It is very resilient, robust, and fault-tolerant. Many industries favor BP neural networks due to the benefits they provide in areas like signal processing, pattern recognition, function approximation, process control, market research, image processing, fault detection, and industry estimates. The typical network structure for a BP neural network integrates three layers: output layer, input layer, and hidden layer. There

is communication between the levels, but none between the nodes on the individual layers. Forward signal transmission and error back propagation are both included into the BP architecture. The two-stage learning method is predicated on the idea that the input vector may be transformed implicitly at a later step in order to derive the output vector, and from this transformation we can infer the mapping connection between the two sets of data. Both forward and backward propagation of input information and output error make up what we call the "information loop" in a BP network. Naturally, this technique requires iteration, which may alter the connections weight co-efficient between every neurons layer, to keep the output error within a predefined range. **Fig. 2** is a flowchart depicting the process of neural network training.

Neural networks possess a high degree of generality and are capable of approximating intricate relationships. Their primary limitation lies in their ability to offer a comprehensive understanding of the relationship's structure, leading to their commonly perceived "black-box" status. The utilization of neural networks necessitates the formulation of numerous modeling assumptions, including the determination of the quantity of hidden layers including the units available in every layer. Normally, there is a shortage of information regarding the appropriate methodology for making these determinations. In addition, it should be noted that the back-propagation algorithm may exhibit slow convergence in the event that an inappropriate learning rate is employed.

The process of reducing the dimensionality of data can be accomplished through the utilization of neural networks. The process of reducing high-dimensional data to low-dimensional codes can be achieved through the utilization of a multi-layered neural network, which is learned with the central layer. This network is designed to restructure high-dimensional input vector. The usage of the gradient descent is applicable for the purpose of refining the weights in autoencoder networks. However, its efficacy is contingent upon the proximity of the initial weights to an optimal solution. A proposition was put forth regarding an efficient method for weight initialization that facilitates the acquisition of low-dimensional codes by deep autoencoder networks. According to Farebrother [26], it has been observed that this tool is more effective in reducing the dimensionality of data as compared to principal components analysis.

Deep Learning

The field of Deep Learning represents a recent development in the realm of machine learning research, aimed at bringing machine learning closer to its initial objective of achieving artificial intelligence. According to Chen, Tang, Xie, Feng, and Zhang [27], Deep Learning pertains to the acquisition of various tiers of representation and abstraction, which facilitate the comprehension of data. Deep learning models have been found to be more efficient in representing specific categories of functions, especially those related to visual recognition. These models are capable of representing more intricate functions using fewer computational resources. Support Vector Machines (SVMs) and the Kernel approaches are not considered to be deep learning techniques. Classification trees lack depth due to the absence of a hierarchical arrangement of features. The utilization of non-convex loss functions is a fundamental aspect of deep learning, and it has been established that deep supervised learning is a non-convex process.

The utilization of deep learning exhibits promising potential in addressing large-scale data, notwithstanding the presence of certain obstacles. Several techniques have been suggested for leveraging unannotated data in deep neural network-driven structures. The techniques employed involve either conducting a layer-wise weights pre-learning in a greedy manner, utilizing solely unlabeled data, and subsequently fine-tuning the model in a supervised manner, or acquiring unsupervised encoding at distinct architecture levels in consideration with supervised signals. In regards to the latter, the fundamental configuration is outlined as follows: The proposed methodology involves the selection of an unsupervised learning algorithm, followed by the selection of models comprising the deep architecture. The unsupervised learning algorithm is then integrated into one or more architecture layers as auxiliary tasks. Finally, both the unsupervised and supervised tasks are trained concurrently utilizing similar architecture.

Comparison of Various Techniques and Ensemble Techniques

Ensemble learning encompasses a variety of techniques, among which boosting, random forest, and bagging are the most frequently employed. The utilization of a bootstrap (or bagged) classifier has been observed to yield superior results when compared to a singular classifier, which is retrieved majorly from original training dataset. The enhanced precision is a result of the composite model’s capability to reduce the constituent classifiers’ variance. The utilization of a bagged predictor has been observed to enhance the accuracy of predictions in comparison to a single predictor. The model exhibits resilience to noisy data and overfitting. Bootstrap approaches can be employed not only for evaluating the deviation of a model, but also for enhancing its precision. Bagging and boosting techniques employ an ensemble of models and aggregate the outcomes of multiple methods. Bagging and boosting are viable techniques for both classification and prediction, as evidenced by Nasir Amin, Iftikhar, Khan, Faisal Javed, Mohammad AbuArab, and Faisal Rehman [28].

The technique known as bagging, short for bootstrap aggregation is a classification method that falls under the category of ensemble learning. It involves generating multiple bootstrap sets, with replacements, from the initial learning dataset to establish a collection of training datasets that exhibit slight variations. Bagging refers to the methodology of gathering a random assortment of observations and placing them into a container. The creation of multiple bags involves the random selection of observations from the training dataset, as stated in [29]. Bagging is a machine learning technique that involves aggregating multiple base learners by training them on different bootstrap samples of the original dataset. The base learners are then combined through a voting mechanism to make predictions. The utilization of a collection of base

learners to enhance prediction accuracy is a characteristic feature of the bagging technique. Boosting is an ensemble technique that aims to enhance the performance of learning algorithms by integrating multiple simpler algorithms. The boosting technique bears resemblance to the bagging approach. The initial step involves the sequential construction of foundational learning, whereby each subsequent learner is developed to predict the residual values of the previous learner. By utilizing a complementary learner, the errors committed by prior learners are employed to instruct subsequent foundational learners.

The process of boosting involves training the base classifiers on distinct samples, as indicated in [30]. The performance of boosting may be compromised in cases where the available data is inadequate or when the weak models exhibit excessive complexity. The phenomenon of noise can also affect the performance of Boosting, as noted in [31]. The AdaBoost algorithm, which is characterized as "adaptive," is widely regarded as the most popular boosting algorithm. According to Adyalam, Rustam, and Pandelaki [32], AdaBoost is a straightforward and easily implementable algorithm, which is comparatively less complex than Support Vector Machines (SVMs). Additionally, AdaBoost has demonstrated high efficacy in producing favorable outcomes. The AdaBoost algorithm is capable of handling both numerical and nominal data types.

The aforementioned method exhibits a low generalization error; presents ease of implementation, is compatible with a wide range of classifiers, and does not require any parameter tuning. Nevertheless, it is susceptible to outliers. Even though randomization and bagging may produce comparable outcomes, there can be advantages to utilizing both techniques in conjunction with one another, as they introduce randomness in distinct and potentially complementary manners. The random forest algorithm, which is widely used for machine learning, generates a random decision tree in every form of iteration of the bagging approach. This approach frequently yields highly effective predictors, as noted in previous research. The method of random forests is an ensemble approach based on trees, which comprises a fusion of numerous models. The aforementioned is a classifier ensemble comprising a multitude of decision trees.

The random forest algorithm generates an ensemble of decision trees, thereby enabling the acquisition of multiple outputs from a single input. The classification or regression output is determined by utilizing the predominant votes from all decision trees. The efficacy of random forest models is typically comparable to that of nonlinear classifiers, including SVM, and ANN. The random forest algorithm is a suitable option for constructing a model due to its minimal data pre-processing requirements, lack of necessity for data standardization, and robustness in handling outliers. The requirement of variable assortment is evaded as the approach proficiently performs the task on its own. Due to the utilization of two tiers of randomness, namely observations and variables, in the construction of numerous trees, each tree can be regarded as an autonomous model.

The random forest algorithm employs the technique of bagging to present randomized samples into the decision tree building process, resulting in the creation of multiple decision trees. Typically, the random forest algorithm refrains from implementing decision tree pruning during the construction of each individual decision tree. Models that are overfitted have a tendency to exhibit poor performance when presented with new data. Nevertheless, it has been observed that a random forest integrating overfitted trees can yield a highly effective model, which exhibits satisfactory performance on novel data.

III. MACHINE LEARNING AND DATA MINING IN BIG DATA ANALYTICS

Hadoop is a software framework utilized for the purpose of conducting Big Data analytics, and it serves as an open-source application of MapReduce. The subsequent concise enumeration delineates the MapReduce realization of three key approaches presented in **Table 2**.

Table 2. Key algorithms in MapReduce

Naïve Bayes	This algorithm is among the few, which can be implemented in MapReduce in a natural manner. The process of calculating sums in MapReduce is straightforward. In the Naïve Bayes method, it is possible to compute the likelihood of a feature for a given class. The outcomes pertaining to a specific class can be assigned to a single mapper, and the summation of these outcomes can be performed using a Reducer.
Support vector machines	The proximal SVM is a computationally efficient variant of the SVM algorithm that can be readily implemented within a MapReduce model.
Single-value decomposition	This is a computationally effective technique for estimating eigenvalues. The present algorithm exhibits suitability for application in a sequence of MapReduce tasks, thereby enabling efficient identification of singular values within a matrix of significant size.

Nevertheless, the aforementioned three techniques are not applicable in the context of Big Data analysis. The utilization of traditional ML technique for the classification of big data is deemed inappropriate due to several reasons. Firstly, an ML approach, which has been learned on a particular labeled data domain or dataset, may not be utilized to another data domain or dataset. Secondly, an ML approach is normally learned using a smaller number of classified types, whereas big data encompasses a vast array of class types that are constantly evolving. Thirdly, an ML technique is designed to cater to a single training task, and therefore, it is not effective for multiple learning tasks as well as the transfer

of knowledge for the analytics of big data. Lastly, the memory constraints pose a vital challenge. Algorithms generally operate under the assumption that training data samples are present in primary memory.

However, this assumption is not applicable to big data, as it exceeds the capacity of primary memory. The task of mining big data presents greater difficulties when compared to conventional data mining algorithms. Using clustering as an illustration, an intuitive approach to clustering large-scale data involves the expansion of current techniques (e.g., k-means) to enable them to handle the substantial workloads. Typically, extensions depend on the analysis of a specific quantity of big data samples, and differ in their approach to utilizing the sample-based outcomes for establishing a divider for the entire dataset.

The k-Nearest Neighbor (k-NN) classification algorithm does not produce a classifier framework in a direct manner. Rather, they retain all training data in the system's memory. Therefore, these entities are not suitable for implementation in big data scenarios. The selection of decision tree splitting criteria is reliant of quality measures like data gain. This necessitates the handling of the complete dataset for each expanding node. The application of decision trees to big data poses a challenge. The Support Vector Machine (SVM) algorithm has demonstrated favorable performance when applied to datasets of moderate size. Big data applications are inherently limited. The utilization of deep machine learning exhibits promising capabilities in addressing large-scale datasets. Nevertheless, big data applications pose certain challenges for this approach due to the substantial amount of time required for training.

The challenges associated with deep learning in the realm of Big Data analytics refers to the incremental training for dataset that are non-stationary, large-scale frameworks, and high-dimensional data, as indicated by Kulkarni, Kumar, and Rao [33]. The Variety feature of the big data analytics refers to the input data diversity of its domains and types within the realm of big data. The phenomenon of domain adaptation in the context of deep learning is a significant area of research, which pertains to situations where the learning dataset and its distribution are distinct to the testing dataset. Particular fields of big data, e.g., cyber privacy, may involve a corpus of data that comprises a combination of labeled and unlabeled data. In instances of this nature, deep learning algorithms have the capacity to integrate semi-supervised training techniques with the aim of establishing standards for effective data representation learning.

Representation learning depicts the class of ML approaches, which allow systems to detect the required representations for feature classification or detection directly from unprocessed data. By enabling machines to learn and subsequently apply a feature to a particular task, the need for manual feature engineering is diminished. Representation learning involves the input of data into a machine, which subsequently learns the representation autonomously. The process of defining data representations of particular features, similarity function, and distance function plays a crucial role in determining a predictive model performance. The representation learning process involves the reduction of high-dimensional data to low-dimensional data, which facilitates the identification of patterns and anomalies, as well as a more comprehensive comprehension of the general behaviors of the data. Representation learning is often motivated by the need for computationally and mathematically convenient input processing in machine learning tasks, particularly in classification. The algorithmic definition of certain qualities of real-life data, such as sensor data, images and videos, has proven to be a challenging task. One potential methodology involves analyzing the data for inherent characteristics or depictions, as opposed to relying on explicit methodologies.

Representation Learning Approaches

The utilization of representation learning is imperative in ensuring that the model delivers consistent and disentangled results, thereby enhancing its precision and efficacy. This section will examine the potential of representation learning to enhance model’s effectiveness over three different learning models, such as unsupervised learning, supervised learning, and semi-supervised learning.

Supervised Learning

When a human oversees the training of a machine learning or deep learning model, the process is called supervised learning. By comparing model output to the truth, the computer attempts to self-correct, and the process of training restructure mapping between output and input. This procedure is continued until a global minimum is found for the optimization function. Overfitting occurs when old data still performs poorly even after the optimization algorithm has reached the global minimum. Although a large quantity of data is not required for supervised learning, the learnt features are required in order to effectively learning mappings to output from the input. When added to a supervised learning system, the learnt traits may boost prediction accuracy by as much as 17 percent. In supervised feature learning, features are learnt from an input dataset that has been labeled. There are several instances of supervised learning, such as multilayer perceptrons, supervised dictionary learning, and supervised neural networks.

Unsupervised Learning

Unsupervised learning represents a ML method that prioritizes the observation itself over the labels, disregarding the latter. Unsupervised learning is not typically employed for the purposes of classification or regression. Rather, its primary function is to reveal latent patterns, partition data into clusters, reduce noise, identify anomalies, and perform data decomposition, among other applications. When dealing with data x, it is imperative to exercise caution in selecting features z to ensure the accuracy of the resulting patterns. The observation has been made that an increase in data does not

necessarily equate to an improvement in representations. It is imperative to exercise caution in the construction of a model that exhibits both flexibility and expressiveness, as this will enable the extracted features to effectively communicate crucial information. Unsupervised feature learning is a technique that involves extracting features from input data that lacks labels. This is achieved through various methods, including but not limited to dictionary learning, autoencoders, matrix factorization, independent component analysis, and different types of clustering. The subsequent section will delve further into the methods and workflow, elucidating the intricacies of how they obtain representations.

Supervised Approaches

Supervised Dictionary Learning

The process of dictionary learning involves generating a collection of representative components, commonly referred to as a dictionary, from the given input data. This enables the representation of each data point as weighted summations of representative elements. The attainment of dictionary weights and items could be attained by limiting the mean representation errors over the input data, while simultaneously applying L1 regularizations to weights. This amounts to representations of every point in data, which shows only limited number of weights without a zero. Supervised dictionary learning is a technique that leverages the underlying structure of input data and associated labels to optimize dictionary elements. The technique utilizes dictionary learning as a means of addressing classification problems through the optimization of dictionary components, data point weights, and classifier variables in accordance with the input data. The creation of minimization issue is presented, whereby the objective functionality integrate various elements, such as representation error, classification error, L1 regularization of weights representation for every point of data to effectively facilitate limited representation of data, and L2 regularization of classification algorithm's settings.

Multi-Layer Perceptron

The perceptron, which is a fundamental neural unit, integrates input data sequences, and respective weights, which are evaluated against a reference standard. The Multi-Layer Perceptron (MLP) represents a form of feed-forward neural network that comprises multiple perceptron unit layers. The MLP architecture comprises three nodes layers: hidden layer, output layer, and the input layer. The Multi-Layer Perceptron (MLP) is often denoted as the standard neural network model due to its fundamental architecture within the ANN field. This concept constitutes a fundamental basis for the concepts of latent variables and feature learning. The main objective of this concept is to determine the required weights and variables, which can efficiently signify the major distribution of the complete dataset. This is done in order to obtain outcomes that are nearly identical to the original data when these variables or weights are applied to unfamiliar data. Artificial neural networks (ANN) aid in the extraction of significant patterns from a particular set of data.

Neural Networks

Neural networks constitute a category of machine learning approaches, which utilize a complex arrangement of interlinked nodes distributed across multiple layers. The system is founded upon the nervous system of animals, wherein the nodes bear semblance to neurons and the edges bear semblance to synapses. The neural network defines algorithmic procedures for transmitting input dataset from the input layer to the output layer within the network, with each connection possessing a corresponding weight. The neural network's network function is characterized by the weights that parameterize the connections between the two layers of input and output. By appropriately defining network functions, it is possible to accomplish diverse learning tasks through the minimization of a cost element over a network element.

Unsupervised Approaches

The knowledge representation process and its acquisition from unannotated datasets are commonly known as unsupervised feature learning. The objective of Unsupervised Representation Learning is often to discover low-dimensional features, which compress underlying structures within the high-dimensional input dataset.

K-Means Clustering

The K-means clustering approach represents a method for vector quantization. The process of K-means clustering involves partitioning an n-dimensional vector set to k subsets or clusters. Each vector in the set is allotted to clusters whose mean is closest to it. Although the problem is computationally NP-hard, it is still being approached using suboptimal greedy techniques. The K-means clustering algorithm partitions an unannotated dataset into k clusters, followed by the extraction of centroid-based features. These attributes can be developed through a multitude of methods. One approach involves augmenting each sample with k binary attribute, where the j^{th} value of the feature matches the nearest k-means trained to samples. The utilization of cluster ranges as a feature may be achieved through the application of a radial basis function post-processing.

Local Linear Embedding

The Locally Linear Embedding (LLE) technique is a non-linear technique that is utilized for generating low-dimensional representations that preserve the neighboring relationships of high-dimensional input data, which may not have any labels associated with them. The primary objective of LLE is to perform dimensionality reduction on data sets by utilizing low-

dimensional point of data, whereas simultaneously preserving certain geometric characteristics of the neighboring points in the initial data set. The LLE process comprises of two primary stages. The initial stage involves a "neighbour-preserving" approach, whereby every input point of data Xi is restructured using the weighted summation of –nearest neighbour points of data.

The optimal weight is deduced via the limitation of the mean-square re-structuring error, which represents the discrepancy between input points of data, including its reconstructions. It is fundamental to note that weights connected to every point of data remains equivalent to 1. The subsequent phase encompasses the process of "dimension reduction," which involves the exploration of vectors within a space of lower dimensions, with the aim of minimizing the error of representation while maintaining the optimum weights realized from the previous step. In the first step, the weights are typically optimized on the basis of their fixed set of data that can be controlled using the least-squares method. In the second step, the optimizations of points with a lower dimension are executed with a fixed weight. This could be controlled using the sparse eigenvalue decomposition method.

Unsupervised Dictionary Mining

Unsupervised dictionary learning is a technique that optimizes dictionary elements without utilizing data labels, but rather relies on the inherent structure of the data. The process of unsupervised dictionary learning involves the acquisition of fundamental functions, or dictionary elements, for the purpose of representing data from input data that lacks labeling. Sparse coding is an instance of this form of learning. In instances where the vocabulary item quantity surpasses the dimensionality of the input dataset, the implementation of sparse coding could facilitate the acquisition of overcomplete dictionaries. The K-SVD algorithm is a computational method designed to acquire a set of dictionary components, which allows efficient and sparse representation.

Deep Architectures Approaches

The hierarchical construction of the biological brain structure, characterized by the stacking of multiple layers of learning nodes, serves as the inspiration for deep learning infrastructure formulated for feature learning. The concept of distributed representation is commonly employed in the construction of such architectures, wherein the generation of observable data is attributed to the interactions among numerous heterogeneous components at multiple levels.

Restricted Boltzmann Machine

Restricted Boltzmann Machines (RBMs) [34] are commonly employed as fundamental components in multilayer learning frameworks. A RBM is a type of bipartite network that is undirected and that comprises a collection of visible variables, binary hidden variables, and edges, which interlink the visible and hidden nodes. This is a specific type of Boltzmann machine that is distinguished by the absence of connections between nodes within the network. In a Restricted Boltzmann Machine (RBM), every connection between nodes is associated with a weight value. The interrelationships and magnitudes of the interconnections establish an energy element, which may be employed to produce joint probability distributions of observable and latent nodes. In the context of unsupervised representation learning, it is possible to conceptualize an RBM as a design that consists of a single layer. The observable variables, specifically, pertain to input dataset, whereas the latent variables recount the feature detectors. The employment of the Hinton's contrastive divergence (CD) methodology enables the training of weights through the maximization of the probability of visible variables.

Autoencoders

It has been discovered that deep network representations exhibit insensitivity towards intricate noise or data conflicts. To a certain degree, this can be correlated with the field of architecture. The utilization of convolutional layers and max-pooling has been demonstrated to yield transformation invariance. Autoencoders present the neural networks, which can be learned to effectively undertake representation learning. Autoencoders purpose to replicate their inputs as outputs through the utilization of an encoder and a decoder. The training of autoencoders is commonly accomplished through the utilization of recirculation, which is a learning mechanism that involves comparing the input network activation to that of their reconstructed inputs.

Representation learning algorithms facilitate the attainment of high classification accuracy through supervised learning techniques while maintaining computational efficiency. The process of transforming data into another domain while maintaining its original characteristics is carried out with the aim of enhancing the accuracy of classification algorithms, reducing computational complexity, and increasing processing speed. The classification of Big Data necessitates the utilization of a multi-domain, representation-learning (MDRL) technique due to the vast and expanding nature of its data domain. The MDRL methodology comprises the acquisition of feature variables, extraction of features, and learning of distance metrics. Numerous representation-learning approaches have been recommended in the domain of ML research.

The cross-domain, representation-learning (CDRL) approach, which has been proposed recently, may be a viable option for big data classification when used in conjunction with the proposed networking model. Nonetheless, the implementation of the CDRL approach to the classification of big data is likely to face several obstacles, such as the challenge of feature selection, geometric representation construction, feature extraction, and data type separation. The presence of a continuity parameter in big data necessitates the implementation of lifelong learning techniques to tackle the

associated challenges. The acquisition of knowledge pertaining to the attributes of big data in a brief period may not be conducive to its retention and applicability over an extended duration. Therefore, it is recommended to employ machine lifelong learning (ML3) techniques. The ML3 framework is a concept that enables the preservation of acquired knowledge from training examples across various learning stages.

IV. CONCLUSION

The utilization of dimensionality reduction techniques can facilitate the process of data visualization. The Principal Component Analysis (PCA) is a frequently implemented approach for minimizing the available number of dimensionality in a given dataset. Factor analysis is a statistical technique that can be employed for the purpose of reducing data or detecting underlying structures within it. The k-means algorithm is deemed to be computationally efficient; however, it may converge to a suboptimal solution. The k-Nearest Neighbor (k-NN) algorithm is characterized by its ease of implementation and resilience to outliers in the predictor variables. Nonetheless, it encounters significant challenges when dealing with data that exhibit mixed types. Support Vector Machines (SVM) demonstrates proficiency in addressing challenges posed by sparse, nonlinear, and high-dimensional problems. However, SVM exhibits limitations in effectively managing computational scalability and mixed data types. Decision trees have demonstrated high performance in handling extensive datasets, although they may result in overfitting. The act of pruning trees is executed with the purpose of eliminating irregularities present in the data used for training, which may be attributed to the presence of outliers and noise. Logistic regression represents a computationally efficient method; however, it is susceptible to underfitting and can possibly present suboptimal precision.

The Naive Bayes algorithm refers to a characterized by its ease of construction and rapid training, rendering it appropriate for relatively modest training sets. However, it is susceptible to bias. Although neural networks exhibit strong predictive capabilities and are capable of handling noisy data, their efficacy in processing large datasets with intricate models is limited. The employment of a combination of models to enhance accuracy is a common practice in ensemble methods, with bagging, boosting, and random forests being the most frequently utilized techniques. Conventional technologies encounter difficulties when processing unstructured and large-scale data sources. BDaaS has the potential to function as an expanded layer within the service structure. Machine learning and conventional data mining approaches, such as, k-NN, k-means, SVM, and decision trees, are not well-suited for managing large-scale datasets. The utilization of deep learning exhibits promising potential in addressing large-scale datasets, notwithstanding the obstacles encountered.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding was received to assist with the preparation of this manuscript.

Ethics Approval and Consent to Participate

The research has consent for Ethical Approval and Consent to participate.

Competing Interests

There are no competing interests.

References

[1]. Y. Tang et al., “Characterization of Calculus bovis by principal component analysis assisted qHNMR profiling to distinguish nefarious frauds,” J. Pharm. Biomed. Anal., vol. 228, no. 115320, p. 115320, 2023.
[2]. Haldorai, A. Ramu, and S. A. R. Khan, Eds., “Business Intelligence for Enterprise Internet of Things,” EAI/Springer Innovations in Communication and Computing, 2020, doi: 10.1007/978-3-030-44407-5.
[3]. Haldorai and U. Kandaswamy, “Intelligent Spectrum Handovers in Cognitive Radio Networks,” EAI/Springer Innovations in Communication and Computing, 2019, doi: 10.1007/978-3-030-15416-5.
[4]. M. Hajjar, G. Aldabbagh, and N. Dimitriou, “Using clustering techniques to improve capacity of LTE networks,” in 2015 21st Asia-Pacific Conference on Communications (APCC), 2015.
[5]. F. D. F. Duarte, “Multimodal optimization with the local optimum ranking 2 algorithm,” Research Square, 2022.
[6]. L. Nigro, “Performance of parallel K-means algorithms in Java,” Algorithms, vol. 15, no. 4, p. 117, 2022.
[7]. Y. Gao, Y. Hu, and Y. Chu, “Ability grouping of elderly individuals based on an improved K-prototypes algorithm,” Math. Probl. Eng., vol. 2023, pp. 1–11, 2023.
[8]. M. A. N. D. Sewwandi, Y. Li, and J. Zhang, “A class-specific feature selection and classification approach using neighborhood rough set and K-nearest neighbor theories,” Appl. Soft Comput., vol. 143, no. 110366, p. 110366, 2023.
[9]. Y. Tang, Y. Chang, and K. Li, “Applications of K-nearest neighbor algorithm in intelligent diagnosis of wind turbine blades damage,” Renew. Energy, vol. 212, pp. 855–864, 2023.
[10]. L. Wang, M. Zhuang, and K. Yuan, “Active control method for rotor eccentric vibration of high-speed motor based on least squares support vector machine,” Machines, vol. 10, no. 11, p. 1094, 2022.
[11]. Y. Feng and Q. Wu, “A statistical learning assessment of Huber regression,” J. Approx. Theory, vol. 273, no. 105660, p. 105660, 2022.

[12]. X. Liu, J. Liu, and X. Chen, "A novel method of identifying optimal interval regression model using structural risk minimization and approximation error minimization," in 2021 33rd Chinese Control and Decision Conference (CCDC), 2021.
[13]. Y. Deng, N. Gazagnadou, J. Hong, M. Mahdavi, and L. Lyu, "On the hardness of robustness transfer: A perspective from Rademacher complexity over symmetric difference hypothesis space," arXiv [cs.LG], 2023.
[14]. V. Grabstaite, R. Baleviciute, R. J. Luiniene, M. Landauskas, and A. Vainoras, "Physiologic changes of ECG parameters in actors during performance – reaction complexity," J. Complex. Health Sci., vol. 3, no. 2, pp. 137–142, 2020.
[15]. V. Vapnik and R. Izmailov, "Rethinking statistical learning theory: learning using statistical invariants," Mach. Learn., vol. 108, no. 3, pp. 381–423, 2019.
[16]. M. Mahsuli and T. Haukaas, "Risk minimization for a portfolio of buildings considering risk aversion," J. Struct. Eng. (N. Y.), vol. 145, no. 2, p. 04018241, 2019.
[17]. K. Ashok, M. Ashraf, J. Thimmia Raja, M. Z. Hussain, D. K. Singh, and A. Haldorai, "Collaborative analysis of audio-visual speech synthesis with sensor measurements for regulating human–robot interaction," International Journal of System Assurance Engineering and Management, Aug. 2022, doi: 10.1007/s13198-022-01709-y.
[18]. T. Bellotti, R. Matousek, and C. Stewart, "A note comparing support vector machines and ordered choice models' predictions of international banks' ratings," Decis. Support Syst., vol. 51, no. 3, pp. 682–687, 2011.
[19]. H and A. R, "Artificial Intelligence and Machine Learning for Enterprise Management," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Nov. 2019, doi: 10.1109/icssit46314.2019.8987964.
[20]. M. V. Da Silva et al., "A data-driven examination of apathy and depressive symptoms in dementia with independent replication," bioRxiv, 2022.
[21]. Haldorai and U. Kandaswamy, "Energy Efficient Network Selection for Cognitive Spectrum Handovers," EAI/Springer Innovations in Communication and Computing, pp. 41–64, 2019, doi: 10.1007/978-3-030-15416-5_3.
[22]. B. M. Greenwell, "Conditional inference trees," in Tree-Based Methods for Statistical Learning in R, Boca Raton: Chapman and Hall/CRC, 2022, pp. 111–146.
[23]. H. Zhang and C. X. Ling, "Geometric properties of naive Bayes in nominal domains," in Machine Learning: ECML 2001, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 587–599.
[24]. G. Zhang, P. Nulty, and D. Lillis, "Enhancing legal argument mining with domain pre-training and neural networks," J. Data Min. Digit. Humanit., vol. NLP4DH, 2022.
[25]. M. Nielsen, L. Wenderoth, T. Sentker, and R. Werner, "Self-supervision for medical image classification: state-of-the-art performance with ~100 labeled training samples per class," arXiv [cs.CV], 2023.
[26]. R. W. Farebrother, "Notes on the prehistory of principal components analysis," J. Multivar. Anal., vol. 188, no. 104814, p. 104814, 2022.
[27]. R. Chen, Y. Tang, Y. Xie, W. Feng, and W. Zhang, "Semisupervised progressive representation learning for deep multiview clustering," IEEE Trans. Neural Netw. Learn. Syst., vol. PP, 2023.
[28]. M. Nasir Amin, B. Iftikhar, K. Khan, M. Faisal Javed, A. Mohammad AbuArab, and M. Faisal Rehman, "Prediction model for rice husk ash concrete using AI approach: Boosting and bagging algorithms," Structures, vol. 50, pp. 745–757, 2023.
[29]. N. S. F. Putri, A. P. Wibawa, H. Ar Rasyid, A. Nafalski, and U. R. Hasyim, "Boosting and bagging classification for computer science journal," Int. J. Adv. Intell. Inform., vol. 9, no. 1, p. 27, 2023.
[30]. M. Zhan, X. Shi, F. Liu, and R. Hu, "IGCNN-FC: Boosting interpretability and generalization of convolutional neural networks for few chest X-rays analysis," Inf. Process. Manag., vol. 60, no. 3, p. 103258, 2023.
[31]. J. Wang, R. Min, Z. Wu, and Y. Hu, "Boosting I/O performance of internet servers with user-level custom file systems," Perform. Eval. Rev., vol. 29, no. 2, pp. 26–31, 2001.
[32]. T. R. Adyalam, Z. Rustam, and J. Pandelaki, "Classification of osteoarthritis disease severity using adaboost support vector machines," J. Phys. Conf. Ser., vol. 1108, p. 012062, 2018.
[33]. A. R. Kulkarni, N. Kumar, and K. R. Rao, "Efficacy of Bluetooth-based data collection for road traffic analysis and visualization using big data analytics," Big Data Min. Anal., vol. 6, no. 2, pp. 139–153, 2023.
[34]. M. Kālis, A. Locāns, R. Šikovs, H. Naseri, and A. Ambainis, "A hybrid quantum-classical approach for inference on restricted Boltzmann machines," arXiv [quant-ph], 2023.