Mesopotamian journal of Big Data
Vol. **(2023)** 2023, **pp**. 1–11
DOI: https://doi.org/10.58496/MJBD/2023/001 ISSN: 2958-6453
https://mesopotamian.press/journals/index.php/BigData

Research Article

Scalable Neural Network Algorithms for High Dimensional Data

Mukesh Soni[1,*], Marwan Ali Shnan [2]

[1]*IEEE Computer Society: Washington D.C., District of Columbia, United States.*

[2]*Faculty of Computer Systems & Software Engineering (FSKKP) UMP, Malaysia*.

ARTICLEINFO

Article History

Received 12 Nov 2022
Accepted 7 Jan 2023
Published 15 Jan 2023

Keywords

Neural Network

AI

Big Data

High dimensional data

Machine Learning.

ABSTRACT

The boundary for machine learning engineers lately has moved from the restricted data to the algorithms' failure to involve every one of the data in the time permitted. Due of this, scientists are presently worried about the adaptability of machine learning algorithms notwithstanding their exactness. The key to success for many computer vision and machine learning challenges is having big training sets. A few published systematic reviews were taken into account in this topic. Recent systematic reviews may include both more recent and older research on the subject under study. Thus, the publications we examined were all recent. The review utilized information that were gathered somewhere in the range of 2010 and 2021. System: In this paper, we make a modified brain organization to eliminate possible components from extremely high layered datasets. Both a totaled level and an exceptionally fine-grained level of translation are feasible for these highlights. It is basically as easy to grasp non-straight connections as it is a direct relapse. We utilize the method on a dataset for item returns in web based shopping that has 15,555 aspects and 5,659,676 all out exchanges. Result and conclusion: We compare 87 various models to show that our approach not only produces higher predicted accuracy than existing techniques, but is also interpretable. The outcomes show that feature selection is a useful strategy for enhancing scalability. The method is sufficiently abstract to be used with many different analytics datasets

1. INTRODUCTION

High dimensionality data sets are where machine learning methods are increasingly used[1]. The majority of algorithms were created when there were fewer and smaller data sets, however today different tradeoffs are needed for limited scope and enormous scope learning tasks. The run of the mill guess assessment compromise applies to limited scope learning issues. The trade-off is especially difficult when it comes to large-scale learning issues because it affects both the algorithm's accuracy and computational complexity[2]. The greater part of algorithms was created with the assumption that the data set will be addressed as a solitary memory-inhabitant table, which further mixtures the issue. In this way, these methods are incapable on the off chance that the total data set can't fit in primary memory. For a variety of reasons, a NN's performance is highly dependent on the architecture selected[3]. First off, a NN's prediction computation is significantly influenced by its architecture. In fact, NNs with various architectures can generate various outputs from the same information. From one perspective, too simple of structures might not have the necessary articulation to recreate complicated relations. This could lead to low variance and under fitted predictions with high bias. On the other hand, over fitting and other numerical artifacts may result from overly complicated designs, producing expectations with low inclination and high change [1]. Second, a NN's topology influences how hard the model is to compute because more layers and nodes require more drifting point tasks to prepare the model and produce forecasts. Deciding a suitable design is in this manner a critical initial step that can fundamentally affect both the computational intricacy expected to prepare a profound learning (DL) model and a definitive prescient force of the DL model itself. In any case, the size of the boundary space for NN models forestalls a careful hunt. In reality, the quantity of geographies increments dramatically as layer thickness, neurons per layer and associations between layers increment. In the age of massive and streaming data, neural network algorithms are well-positioned to serve as the foundation of machine learning applications thanks to all these properties. Using picked Kohonen neurons (centers) from various Kohonen nets, we present a smart brain network learning

*Corresponding author. Email: msoni10@gmail.com

technique in this paper that: (1) can be parallelized at various levels of granularity; (2) deals with the issue of high-layered information through class-based feature assurance; (3) learns a social event of classifiers; and (4) is not difficult to execute on hardware. We train a lot of Kohonen nets in equivalent using streaming information to make a couple of representative information centers for dimensionality decline through feature decision [2].

2. OVERVIEW OF NEURAL NETWORK ALGORITHMS

Profound neural networks (NN) are nonlinear models that utilize observational data to surmised obscure capabilities. Their muddled construction, which empowers these strategies to reproduce unpredictable connections between amounts picked as information sources and results of the model, is the wellspring of their expansive materialness. A NN can be considered numerically as a coordinated non-cyclic chart with layers of hubs (otherwise called neurons). For the NN to precisely address convoluted elements among information sources and results, the sort of network between the different layers is pivotal. The quantity of layers, the quantity of hubs at each layer, and the association between hubs in adjoining layers act as an overall outline of the chart's construction or plan [3].

For a variety of reasons, a NN's performance is highly dependent on the architecture selected. First off, a NN's prediction computation is significantly influenced by its architecture. In fact, NNs with various architectures can generate various outputs from the same input. From one perspective, too simple of structures might not have the necessary articulation to recreate complicated relations. As a result, the predictions may not adequately fit the data, having a large inclination and low change. Then again, over fitting and other mathematical relics might result from excessively confounded plans, delivering forecasts with low inclination and high change. Second, a NN's geography impacts the way in which hard the model is to figure since additional layers and hubs require really drifting point tasks to prepare the model and produce forecasts. Deciding a suitable design is hence an essential initial step that can fundamentally affect both the computational intricacy expected to prepare a profound learning (DL) model and a definitive prescient force of the DL model itself. In any case, the size of the boundary space for NN structures forestalls an exhaustive hunt. In reality, the quantity of geographies increments dramatically as layer thickness, neurons per layer and associations between layers increment [4].

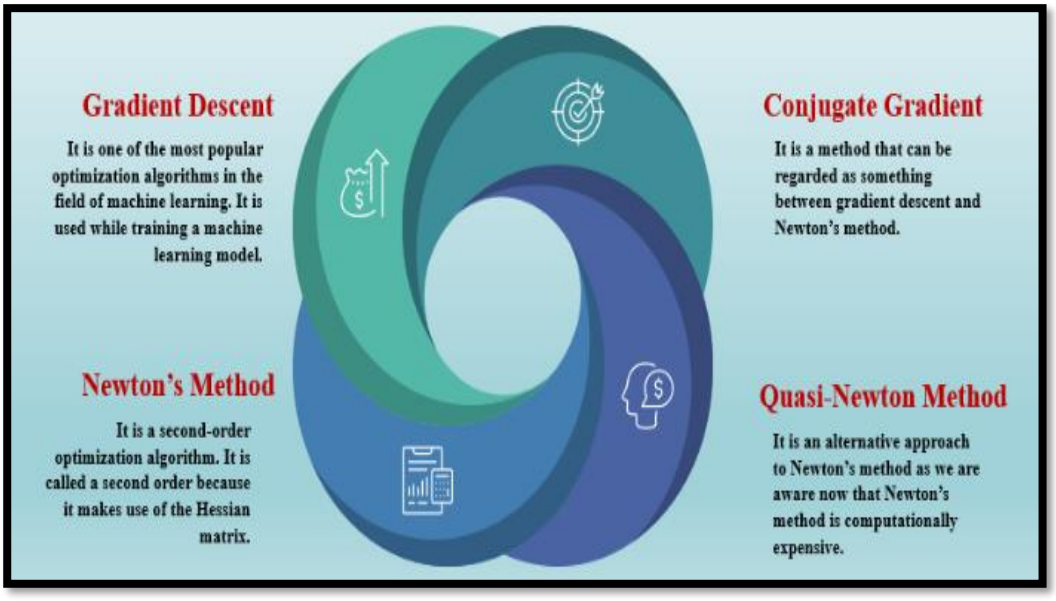

Fig. 1 Neural Network Algorithm

2.1 Gradient Descent

One of the most well-liked optimization techniques in machine learning is this one. When a machine learning model is being trained, it is employed. In plain English, it is mostly used to identify coefficient values that merely minimize the cost function. To decrease the lost capability, we initially characterize specific boundary values and afterward, utilizing

math, start to iteratively change the qualities. Let's get to the section about what a gradient is now. An inclination, frequently known as the slant, means how much the result of any capability will change in the event that the info is marginally diminished. Similar to how a steep slope causes a model to learn more quickly, a zero slope causes a model to stop learning. This is so that the specified algorithm can be minimized using the minimization algorithm [5]. The equation for deciding the accompanying situation on account of inclination decline is introduced underneath.

$$b = a - Y\nabla f(a) \quad (1)$$

2.2 Newton's Method

It is an algorithm for second-request streamlining. It uses the Hessian matrix, which is why it is referred to as second-order. A squared framework of second-request halfway subordinates of a scalar-esteemed capability is all that the Hessian grid is, at the end of the day. To find the roots or fixed recognizes, the Newton's strategy improvement algorithm is utilized to the principal subsidiary of a twofold differentiable capability f. Let's now discuss the processes needed to optimize using Newton's approach.

It starts by assessing the loss index. The halting criteria are then verified as true or incorrect. If this is the case, it determines Newton's training direction and rate, improves the neuron's parameters or weights, and repeats the cycle. As a result, you can now claim that it requires less steps than gradient descent to arrive at the function's minimal value. Although it requires less steps than the gradient descent approach, it is still not generally utilized since the accurate calculation of the hessian and its inverse requires a lot of processing power [6].

2.3 Conjugate Gradient

It is a technique that falls between Newton's method and gradient descent. The primary distinction is that it quickens the slow convergence that is often associated with gradient descent. It is an iterative algorithm, which is another crucial element. It may be applied to both linear and non-linear systems. Magnus Hestenes and Eduard Stiefel created it. As was already noted, the Conjugate Gradient algorithm yields faster convergence than gradient descent; the reason for this is that the search is carried out simultaneously with the conjugate directions, which causes it to converge more quickly than gradient descent approaches. It's vital to keep in mind that is referred to as the conjugate parameter. Intermittently, the preparation course is reset to the angle's negative bearing. This approach is better than inclination drop for preparing neural networks because it avoids the Hessian matrix, which adds to the computing overhead, and because it converges more quickly. It is suitable for usage in substantial neural networks [7].

2.4 Quasi-Newton Method

It is an elective system to Newton's strategy since, as we presently know, Newton's method requires a lot of calculation. These issues are somewhat addressed by this method, which creates an estimate of the inverse Hessian at each iteration of the algorithm rather than first computing the Hessian matrix and then, at that point, the backwards straightforwardly. Now, this approximation is computed using data from the loss function's first derivative. We may therefore conclude that because it reduces calculation time and is considerably quicker than gradient descent or conjugate gradient method, it is arguably the method best suited to handle massive networks [8].

3. LITERATURE REVIEW

Hancock & Khoshgoftaar have done a thorough examination of typical methods for pre-processing categorical data for deep neural networks. On a few even informational indexes and different profound learning designs, the creators looked at existing unmitigated information encoding procedures[9].

Kuzey et al. (2014) examined artificial neural networks and decision trees as two machine learning techniques for determining the relative weights of elements as predictors of company value. They positioned their importance utilizing a responsiveness investigation in view of data combination, utilizing multi-nationality (estimated by the worldwide deals proportion) and fourteen other monetary pointers on organization esteem as information factors. Their study demonstrates that both strategies produced a comparable set of significant predictors, but more research is still needed to determine how accurate these methods are when dealing with high-dimensional input spaces [10].

One of the main frightening little creature vectors of a couple of sicknesses is the mosquito, as per Keun Youthful Lee et al (2016) .'s concentrate on the benefit of utilizing ANN in mosquito species. Especially as of late, as the quantity of individuals who appreciate outside exercises in metropolitan regions has expanded because of the mosquito. Furthermore, expecting mosquito action is pivotal for dealing with individuals' wellbeing and strength[11].

According to Sucheta V. Kolekar and colleagues (2010), the majority of e-learning frameworks fall short when it comes to taking the needs and learning preferences of the end user into account. This is because they are primarily focused on the online content era. This demonstrates the need for adjusting for the client's learning. The term "adaptive e-learning" refers to a teaching framework that takes academic viewpoints into account while understanding the learning content and user interface [12]. Palukuru NAGENDRA et al. (2010) examined how to figure out the movement capabilities in a multi-territory management framework using an artificial neural system (ANN) based model and a multilayer perceptron (MLP) arrange. The stack status is the ANN's contribution, and the yields are the exchange capacity between framework territories, voltage extends, and voltage points at concerned transports in the areas under consideration [13].

A three-layered architecture concept for big data storage and analysis was presented by M. U. Bokhari et al. Data collection, data storage, and data analysis and report generating comprise the three layers. A group of high speed hubs or servers are kept on the data gathering layer to gather and deal with the tremendous number of enormous data coming from quick sources, similar to sensors or virtual entertainment. The huge data must be kept in the data storage layer [14].J.L. Berral-Garcia listed the most popular machine learning techniques for big data analytics. To conduct modeling, prediction, and clustering tasks, many algorithms are utilized. The algorithms given in this paper incorporate choice tree algorithms, (for example, Truck, Recursive Segment Trees, or M5), K-Closest Neighbor algorithms, Bayesian algorithms (utilizing Byes hypothesis), Backing vector machines (SVM), Counterfeit Neural Network, K-implies, and DBSCAN algorithms. Map-Lessen Systems (Apache Hadoop and Flash), Google's Tensor Stream, and Microsoft's Purplish blue ML were among the execution structures highlighted. People in general can get to the executions of the recently expressed algorithms through various apparatuses, stages, and libraries, including Rcran, Python Sci-Unit, Weka, MOA, Versatile Pursuit, Kibana, and so forth [15].

4. PROPOSED METHODOLOGY

4.1 Separability Index of Features, Dimensionality Reduction

Gaining from high-layered information is one of AI's key troubles. For high-layered streaming information, a couple of new techniques have been made for both online incorporate assurance and part extraction. Not even one of them, in any case, are for picking highlights well defined for a class. Roy has proposed approaches that utilization a part of the first elements in class-explicit classifiers starting around 1997 at different meetings, and Roy et al. (2013) gives one such technique. The Roy et al. (2013) procedure doesn't, nonetheless, apply to streaming data. Through the course of class-explicit component determination, we distinguish unmistakable capabilities for each class that are the best at separating it from different classes. The thought behind our methodology is equivalent to that of the Most extreme Edge Basis (MMC) and LDA approaches, which amplify between-class dissipate and decrease inside class disperse. All in all, those procedures plan to augment the distance between particular class communities while likewise limiting the distance between data guides having a place toward a similar class. Albeit not an element extraction technique, our methodology is based on a similar thought.

It is very easy to pick highlights in a disconnected mode where an assortment of data focuses is accessible that increment the typical distance of data points of one class from different classes while at the same time limiting the normal distance of data focuses inside that class. This is the basis on which Roy et al. (2013) rank and choose features, and computer studies indicate that it performs admirably. That approach, however, is inapplicable to streaming data in which no data is kept. The premise for feature selection is the same in the suggested approach, but several Kohonen nets are trained from streaming data instead. We really produce a few delegate data focuses for each class via preparing various Kohonen nets, which is the way we get around the issue of not approaching an assortment of data focuses. The Roy et al. proposed class-based feature assurance procedure is not difficult to utilize once we have a lot of representative information centers (tended to by unambiguous Kohonen neurons in the Kohonen nets). On a conveyed handling stage, we train different Kohonen nets at the same time, with various system sizes and part subsets. Our circulated figuring stage is Apache Flash (2015), albeit other practically identical frameworks can be utilized. The bunch places of a Kohonen net, which are the dynamic hubs or neurons of the Kohonen net, are undifferentiated from delegate tests of the streaming data. We then, at that point, pick highlights by class utilizing these common models [16].

4.2 A Customized Deep Neural Network Architecture

We adjust a profound neural network for this utilization case in view of the numerical thought of AND-Entryways and weight sharing examined in the Estimation segment and considering the data structure examined in the Data Model area.

The profound neural network utilizes hubs with a strategic enactment capability, which is equivalent to the combined dissemination capability of the calculated dispersion, AND-Entryways, and flawed ReLU initiation capabilities for the network's order hubs, as well as numeric and meager info data. Furthermore Entryways are executed as clear weightless pairwise increases, as discussed in the subsection Theoretical Justification. The following definitions apply to leaky ReLU activation functions:

$$y = W^T X + b, if\ y = W^T X + b > 0 \quad (2)$$
$$y = (W^T X + b) * 0.01, if\ y = W^T X + b \leq 0 \quad (3)$$

They are particularly beneficial for deep neural network topologies, as can be demonstrated theoretically and practically.

The suggested neural network architecture contains a total of seven credit assignment paths. What's more Entryways don't combine with the credit task way of a neural network since the credit task not entirely set in stone by the quantity of layers with teachable loads that is gone through to create an expectation. Through pairwise communications on the eight item credits, we can distinguish collaborations between an item and different things in a similar bushel. In hubs 84-87, cross associations between these recovered elements are recorded.

We erase hubs 86 to 90 (the order segment) while separating highlights from this plan and use hubs 64 to 85 as our new result hubs, every one of which has an element of only one neuron for each hub. 24 extricated highlights are created therefore, every one of which relates to a meager variable. This technique is legitimate by the way that the result hub 88's forecast depends on these elements, which suggests that these highlights ought to be prescient when utilized in an alternate grouping algorithm [17].

4.3 Benchmarking using Multilayer Perceptron

We compare the neural network to more conventional methods, specifically the usual feed forward multilayer perceptrons: The proposed design has a credit task way of seven, removes 22 highlights after the subsequent hub, and has a component of ten neurons for every hub for with or without hubs from the subsequent hub and the result hub. To guarantee most extreme likeness, we make a direct feed-forward engineering. Two particular enactment capabilities are tried: The other multi-facet perceptron involves broken ReLU initiation capabilities for all hubs however the result hub (in light of the fact that cracked ReLU can't be perused in a probabilistic manner), while one multi-facet perceptron is completely made out of strategic actuation capabilities. Multilayer perceptron, MLPLog, or MLPReLU will allude to the plans utilized for benchmarking, while neural network or NeuralNet will only be used to distinguish it from the neural network [18].

4.4 Application and Instruction

We provide an interface for Python using SWIG and execute the algorithms in C++ utilizing CUDA. An exclusive system for GPU writing computer programs is called CUDA. We can easily build the neural network by integrating the majority of its creation into loops using a programming language like Python. We have made our code available as an open-source Python library for the use of other researchers.Adam is used to train all three architectures (the leaky ReLU multilayer perceptron and the logistic multilayer perceptron) for 50 ages with a minibatch size of 200 [19].

4.5 Evaluation

This project aims to show an interpretable, customizable neural network. This suggests that the evaluation of our strategy should concentrate on two factors: The first is how well customised neural networks compare to existing methods for dimensionality reduction and classification (predictive accuracy). Second, the knowledge we can derive from these methods (interpretability). The assessment of our model takes into account these two fundamental requirements.

We start by comparing the modified neural networks' predicting performance to that of existing techniques. To do this, we divide our dataset into three sections: the testing set (748,427), the validation set (748,057 samples), and the training set (3,204,058 samples) (751,235 samples). All transactions associated with one basket must be included in the same collection, which is how the dataset is divided by basket IDs. We utilize the endorsement set to refine the hyper limits for our classifiers resulting to setting them up on the arrangement information. After the classifiers have been tuned, we test our dataset against the testing set. No further changes of any sort to either the dimensionality decline or the portrayal procedures are permitted at whatever point we have delivered the disclosures for the testing set, which are then quickly given in the results section of this audit. This cycle guarantees that our results are really out-of-test and are not by any stretch pointlessly fit to the testing set. By turning out a portion of the outcomes we had the option to pull from the model, we then evaluate the interpretability of our philosophy [20].

5. FINDINGS OF THE STUDY

5.1 Kohonen Nets for Arrangement from Streaming Data

Step: 1.Set bucket number j to 0 at the beginning.
Step: 2. To build the j^{th} bucket, increase bucket number j (j = j + 1) and add a few more highly scored characteristics by class k to the $(j-1)^{th}$ bucket.
Step: 3.in a conveyed figuring framework, haphazardly initialise last Kohonen nets with shifting network sizes for each class k and j of list of capabilities. Return to stage 2 and make extra Kohonen nets for the extra element cans in the event that the excess positioned highlights have files bigger than 1.In the event that not, continue to stage 4.
Step: 4.Utilizing streaming data and proper info design determinations in light of the component subset that relates to every individual Kohonen net, train all Kohonen nets all the while. At the point when all of the Kohonen nets join, stop preparing.
Step: 5. to choose the game plan of dynamic centers (neurons) for each class k and every bucket j, process extra streaming information through the reasonable out Kohonen nets without changing the heaps. Besides, get the unique center class counts, and stop when the class rates are unsurprising across all powerful center points.
Step: 6.assuming both the outright class count and the class rate for the larger part class are more than the insignificant edges, a functioning hub ought to be allocated to the greater part class.
Step: 7.to work out the sweep of every dynamic hub, process a few additional streaming data, when the widths or radii are steady, stop.
Step: 8.Keep simply the dynamic hubs that meet the edges and eliminate each and every hub from each Kohonen net.

5.2 Predictive Precision

For feature extraction, we employ the deep multilayer perceptrons and the deep neural network. Nodes 86 to 90 are not used in this process; instead, only nodes 26 to 85 are used. By considering nodes 64 through 85, which all have a dimension of one by design, as our new output nodes using this method, we are able to extract 24 features. We supplement the 20 manually created numeric variables with the retrieved features. Thus, a total of 42 features are produced

TABLE I. PROBABILISTIC OUT-OF-TEST FORECASTS AND GENUINE CLASS NAMES ARE CORRELATED USING PEARSON'S R

	AdaBoost	CART	ERT	GB	LDA	LR	RF	SVM-LIN	SVM-RBF
NeuralNet	0.445	0.414	0.436	0.435	0.405	0.407	0.445	0.349	0.426
MLPLog	0.433	0.412	0.426	0.433	0.404	0.408	0.437	0.354	0.422
MLPReLU	0.433	0.407	0.428	0.431	0.398	0.405	0.437	0.358	0.424
NMF	0.414	0.379	0.393	0.400	0.338	0.348	0.413	0.288	0.376
PCA	0.412	0.372	0.387	0.397	0.338	0.349	0.410	0.313	0.374
LDA	0.404	0.391	0.383	0.400	0.349	0.358	0.410	0.315	0.378
ChiSelect	0.411	0.386	0.386	0.396	0.330	0.344	0.408	0.310	0.373
RanProj	0.407	0.367	0.378	0.391	0.329	0.341	0.402	0.304	0.368
NoSparse	0.391	0.379	0.368	0.383	0.309	0.322	0.393	0.287	0.348

TABLE II. AUC FOR EXTRAPOLATED RESULTS

	AdaBoost	CART	ERT	GB	LDA	LR	RF	SVM-LIN	SVM-RBF
NeuralNet	0.769	0.747	0.760	0.759	0.743	0.746	0.765	0.686	0.754
MLPLog	0.762	0.746	0.753	0.758	0.741	0.744	0.759	0.687	0.751
MLPReLU	0.762	0.742	0.755	0.757	0.742	0.746	0.760	0.680	0.753
NMF	0.748	0.726	0.735	0.738	0.712	0.719	0.746	0.655	0.726
PCA	0.748	0.722	0.731	0.737	0.712	0.719	0.744	0.668	0.726
LDA	0.743	0.732	0.729	0.738	0.715	0.721	0.744	0.669	0.727
ChiSelect	0.747	0.730	0.731	0.736	0.707	0.716	0.743	0.661	0.725
RanProj	0.744	0.716	0.726	0.733	0.705	0.713	0.740	0.664	0.722
NoSparse	0.734	0.726	0.721	0.729	0.694	0.703	0.734	0.655	0.711

We utilize nine unique grouping strategies, which are all drawn from the Scikit-Learn machine learning bundle. We also take structural equation models into consideration while benchmarking, but we discover that scalable standard implementations like SmartPLS are insufficient for a dataset this size and intricacy.

TABLE III. PERFORMANCE OF CLASSIFIERS TRAINED ON ORIGINAL DATA IN PREDICTING

	Pearson's r	AUC
NeuralNet	0.439	0.761
MLPLog	0.435	0.758
MLPReLU	0.429	0.758
LR	0.405	0.752
SVM-LIN	0.378	0.745

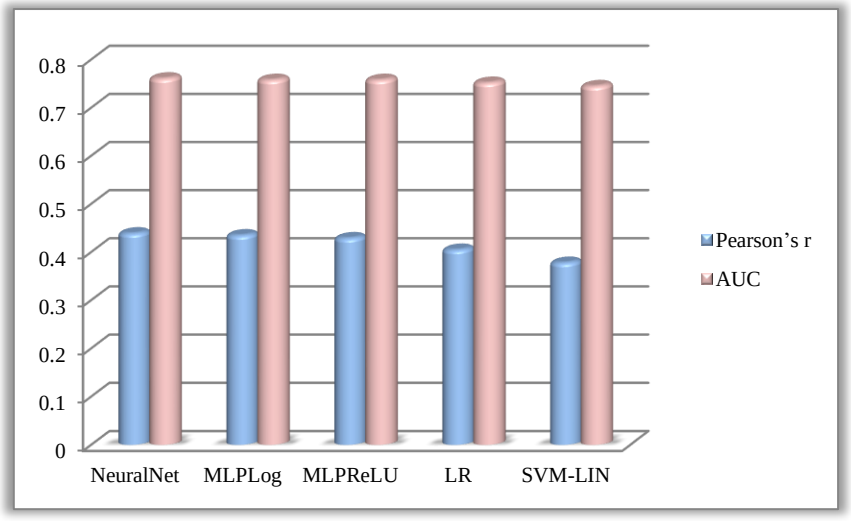

Fig. 2 Graphical representation of performance of classifiers trained on original data in predicting

TABLE IV. THE STATISTICAL SIGNIFICANCE (P-VALUE) OF THE CLASSIFIER'S SUPERIOR PERFORMANCE

	AdaBoost	CART	ERT	GB	LDA	LR	RF	SVM-LIN	SVM-RBF
NeuralNet	-	-	-	-	-	0.354	-	0.273	-
MLPLog	0.000	0.473	0.000	0.310	0.996	-	0.000	0.787	0.005
MLPReLU	0.000	0.000	0.000	0.03	0.000	0.015	0.000	-	0.131
NMF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
PCA	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LDA	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ChiSelect	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RanProj	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NoSparse	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

For features extraction, we combine deep multilayer perceptrons with five popular dimensionality reduction strategies. To ensure most extreme likeness, we utilize these techniques to remove 24 elements, except for direct discriminant investigation, where this is hypothetically inconceivable. We can't, deduced, preclude the likelihood that the meager factors have definitely no prescient importance. Thusly, as an extra benchmark "include extraction procedure" known as NoSparse, we train all classifiers solely on the numeric elements. Where potential, we look at the presentation of profound neural networks and profound multi-facet perceptrons to that of regular grouping strategies since they are sufficiently versatile to be prepared on the full dataset (in particular calculated relapse and a direct portion support vector machine).

With 10 classifiers, 8 part extraction calculations, including NoSparse, and 5 calculations arranged on the first dataset without dimensionality decline, an amount of 85 specific models are made. We down model our planning set for the extended reason capacity support vector machines to 1% of the principal readiness set size since non-straight part SVMs ought to be ready in twofold design, which has a period complexity of more than O(n2) in how much models. We evaluate the models on the testing set subsequent to adjusting the hyper limits for all calculations on the endorsement set.To accomplish a genuinely out-of-test assessment, the discoveries for the testing set are quickly revealed subsequent to being gotten with no extra model changes. Between the probabilistic conjectures and the genuine class marks on the test set, we process Pearson's r and the AUC score. Since Pearson's r is easy to comprehend and a critical sign of a model's impact size, it was picked as the favored measurement.

The way that various data researchers as often as possible use the AUC (region under bend) score prompted the choice to utilize it. It computes the beneficiary working trademark (ROC) bend's region under the bend. The ROC bend shows the compromise between the genuine positive rate (the extent of things out of undeniably returned things that were accurately anticipated to be returned) and the bogus positive rate (the extent of things out of completely returned things that were inaccurately projected to be returned). An AUC score of 1 would indicate a flawless model. AUC would be close to 0.5 for a random guess. Tables 1, 2, and 3 present the findings.

According to our research, utilising an ensemble learner like AdaBoost or random forest to extract features from deep multilayer perceptrons or neural networks improves prediction accuracy compared to using simply neural networks alone. The deep neural network and AdaBoost combination is the best in general model. The deep neural network and random forest model together are the second-best model. We likewise decide the measurable edge at which, given a particular characterization strategy, the best component extraction methodologies beat other element extraction procedures. To decide if the varieties in Pearson's r are genuinely huge, we use a t-test. The results are displayed in Table 4.

We track down that for the three best gathering strategies (which are likely going to be used in a genuine application), explicitly AdaBoost, unpredictable woodlands, and extremely randomized woods, the outperformance of significant brain network is truly profoundly immense. We furthermore find that no component extractor anytime beats the significant brain network at really basic levels. This leads us to the end that, for this specific case, the proposed feature extraction procedure is in some action as astounding as the bleeding edge and, truly, is overwhelming for classifiers with higher execution.We use exactness, accuracy, and review as extra measurements to affirm these discoveries.The level of precise estimates among all spread the word about expectations is as exactness. Out of the relative multitude of times the model predicts a thing will be returned, accuracy counts how frequently it has been appropriately anticipated that a thing in a bin will be returned. Review includes the cases where, out of the relative multitude of things returned, it has been appropriately guessed that a specific thing in a bushel will be returned.

Accuracy and review are regularly compromised in true models. The exactness, accuracy, and review of the model will be influenced by this decision. Subsequently, to help planners in pursuing a savvy decision, we should evaluate the compromises gave by the different models connection to these measurements

5.3 Interpretability

The neural network also has the advantage of being interpretable in addition to having greater predicting accuracy. The ability to retrieve useful information from its specialised architecture is demonstrated in this section. We can perform a more thorough analysis of the basket interaction variables using the neural network's particular architecture. We concentrate on the most significant container communication for the item subcategory due to space constraints.

The outcomes help us develop a more precise knowledge of how various subcategories interact with one another. The greater the effect on the probability of a return, the higher the weight given to the thing. Albeit both subcategories are excess, the advancement algorithm has had the option to similarly perceive this and treats them. The classifications "weave scarf" and "downy scarf" are another delineation. The risk that either knitted or fleece scarves will be returned rises if a buyer places both of them in the same shopping cart. Naturally, the same applies to a knitted scarf as well as every other item in the table. In summary, we can separate interpretable data in regards to muddled, non-straight cooperation from this very high-dimensional dataset utilizing the neural network design.

5.4 Kohonen neuron ensemble classifier system experimental analysis

Here, we give trial discoveries to the Kohonen outfit algorithm, which involves preparing a gathering of Kohonen neurons for characterization utilizing the chose attributes and choosing highlights specific to each class. In Apache Spark's machine learning package MLlib, we evaluate the algorithm's performance with those of other classification algorithms (2015). We similarly randomly divided the data into training and test sets for Spark testing, then repeated 50 times. The standard deviations and average error rates for various algorithms in Spark's MLlib are shown in Table 5. The classifiers for SVMwithSGD and LogRegWithSGD are trained using the stochastic gradient descent (SGD) technique. Without making any feature selections, all of these algorithms were employed with their default settings. The results for the multiclass problems SRBCT, Lymphoma, and Brain are not in the table because SVMwithSGD and LogRegWithSGD only function for two class issues.

TABLE V. STANDARD DEVIATIONS AND AVERAGE TEST ERROR RATES FOR DIFFERENT APACHE SPARK MLLIB CLASSIFICATION ALGORITHMS

	Kohonen ensemble	**SVM with SGD**	**Naive Bayes**	**LogReg With SGD**	**Random Forest**
Leukemia (AML-ALL)	2.16 (4.98)	5.6 (8.2)	11.28 (9.2)	11.31 (14)	13.10 (12)
Central Nervous System	29(17.49)	34.27 (16)	43.86 (13)	37.27 (15)	42.28 (13)
Colon Tumour	11.69(10.50)	18.58 (13)	9.35 (14)	14 (14)	19.69 (14)
SRBCT	1.69 (4.35)	-	8.35 (14)	-	22.35 (20)
Lymphoma	1.69 (4.35)	-	2.11 (5)	-	6.69 (12)
Prostrate	6.8 (8.14)	14.66 (7)	37.99 (14)	14.6 (12)	20.8 (15)
Brain	21.2 (12.20)	-	20 (21)	-	48.7 (27)

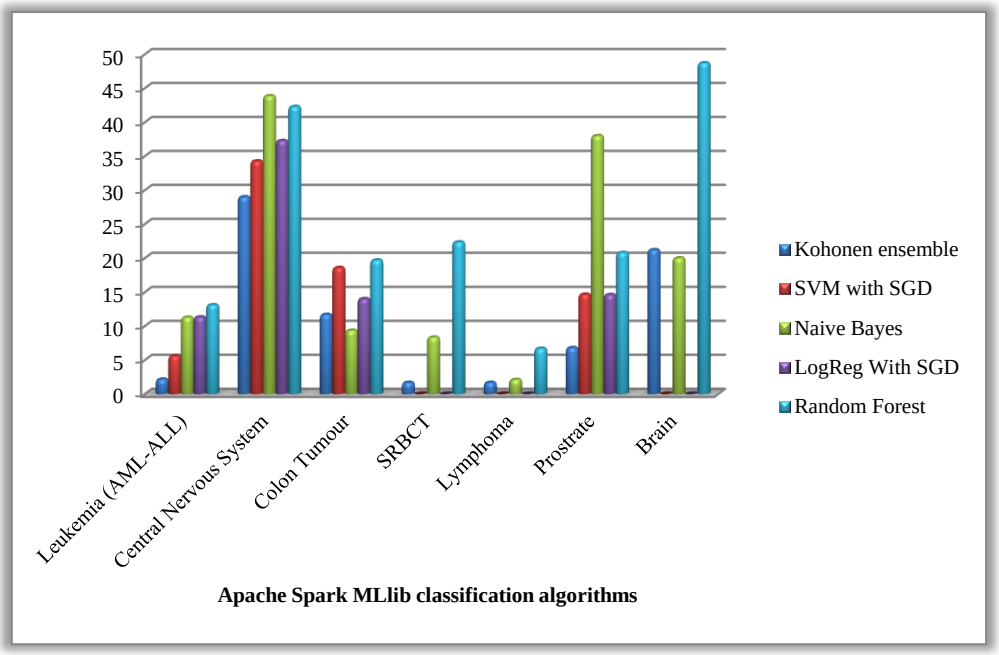

Fig. 1 Graphical representation of Standard deviations and average test error rates for different Apache Spark MLlib classification algorithms

6. CONCLUSION

Making a strategy for extricating interpretable elements from incredibly high dimensional datasets was the point of this review. In reality, the techniques suggested in this study's prediction accuracy improvement have significant practical implications. The mix of AdaBoost and the neural network could deliver a review of more than 35% if we somehow managed to make a framework that mediates on the off chance that the chance of an item being returned surpasses 90% (at the end of the day, make a forecast framework that has an accuracy of generally 90%). The best-performing model simply considering procedure did in scikit-learn, see Tables 1 and 2, would get a survey of negligible over 15%, while the significant multi-layer perceptrons would achieve a survey of practically 30%. A brain gear execution would be undeniably appropriate for Internet of Things (IoT) applications where extraordinarily quick learning and responsiveness are expected for rapid streaming information. Besides, a brain gear execution can deal with saved information by and large quickly. Our preliminary outcomes similarly show that the strategy is particularly perfect at decreasing the parts of a high-layered issue.

Conflicts of Interest

The paper states that there are no personal, financial, or professional conflicts of interest.

Acknowledgment

The authors would like to thank UMP, Malaysia, and the Iraqi Ministry of Education for their support in the conduct of the work published in this paper.

Funding

The absence of funding details in the author's paper suggests that the research was entirely self-funded.

References

[1] R. Agarwal and V. Dhar, "Editorial— Big data, data science, and analytics: The opportunity and challenge for IS research," Information Systems Research, vol. 25, no. 3, pp. 443–448, 2014.

[2] C. Ding, T.-Y. Bao, and H.-L. Huang, "Quantum-inspired support vector machine," IEEE Transactions on Neural Networks and Learning Systems, 2021.

[3] L. Deng, D. Yu, et al., "Deep learning: Methods and applications," Foundations and Trends® in Signal Processing, vol. 7, no. 3–4, pp. 197–387, 2014.

[4] E. H. Houssein, Z. Abohashima, M. Elhoseny, and W. M. Mohamed, "Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision," Expert Systems with Applications, p. 116512, 2022.

[5] S. Furber, D. Lester, L. Plana, J. Garside, E. Painkras, S. Temple, and A. Brown, "Overview of the SpiNNaker system architecture," IEEE Transactions on Computers, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.

[6] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li, "Scalable kNN graph construction for visual descriptors," in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2012, pp. 1106–1113.

[7] M. Muja, "Scalable nearest neighbour methods for high dimensional data," Ph.D. dissertation, Comput. Sci. Dept., Univ. British Columbia, Vancouver, BC, Canada, 2013.

[8] S. C. Hoi, J. Wang, P. Zhao, and R. Jin, "Online feature selection for mining big data," in Proc. 1st international workshop on big data, streams and heterogeneous source mining: Algorithms, systems, programming models and applications, 2012, pp. 93–100.

[9] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," Journal of Big Data, vol. 7, pp. 1–41, 2020.

[10] C. Kuzey, A. Uyar, and D. Delen, "The impact of multinationality on firm value: A comparative analysis of machine learning techniques," Decision Support Systems, vol. 59, no. 1, pp. 127–142, 2014.

[11] K. Y. Lee, N. Chung, and S. Hwang, "Application of an artificial neural network (ANN) model for predicting mosquito abundances in urban areas," Ecological Informatics, vol. 36, pp. 172–180, 2016.

[12] S. V. Kolekar, "Learning Style Recognition using Artificial Neural Network for Adaptive User Interface in E-learning," in Proc. IEEE, 2010, pp. 119–128.

[13] P. Nagendra, S. Halder nee DEY, and T. Dutta, "Artificial Neural Network Application for Power Transfer Capability and Voltage Calculations in Multi-Area Power System," Leonardo Electronic Journal of Practices and Technologies, Issue 16, January-June 2010, pp. 119-128.

[14] M. U. Bokhari, M. Zeyauddin, and M. A. Siddiqui, "An effective model for big data analytics," in Proc. 3rd International Conference on Computing for Sustainable Global Development, 2016, pp. 3980-3982.

[15] J. L. Berral-Garcia, "A quick view on current techniques and machine learning algorithms for big data analytics," in Proc. 18th International Conf. on Transparent Optical Networks, 2016, pp. 1-4.

[16] A. Roy, P. Mackin, and S. Mukhopadhyay, "Methods for Pattern Selection, Class-specific Feature Selection and Classification for Automated Learning," Neural Networks, vol. 41, pp. 113-129, 2013.

[17] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484–489, 2016.

[18] T. Haug, C. N. Self, and M. Kim, "Large-scale quantum machine learning," arXiv preprint arXiv:2108.01039, 2021.

[19] Z. Abohashima, M. Elhosen, E. H. Houssein, and W. M. Mohamed, "Classification with quantum machine learning: A survey," arXiv preprint arXiv:2006.12270, 2020.

[20] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in Neural Information Processing Systems, 2015.