ACTA SCIENTIFIC COMPUTER SCIENCES

Volume 4 Issue 6 June 2022

Research Article

Reduction of Training Data from Large Datasets using Encoder and Decoder Algorithm without Loss of Accuracy

Bagesh Kumar*, Akhil Shukla, Akhil Singh, Mohd Javed Ali and OP Vyas

IIIT Allahabad, India

***Corresponding Author:** Bagesh Kumar, IIIT Allahabad, India.

Received: February 07, 2022

Published: May 25, 2022

Abstract

The objective of this paper is to minimize the number of samples required for training algorithms involving support vectors while maximizing knowledge of the target class. A method is proposed which uses autoencoder in conjunction with farthest boundary point extraction for selecting most promising frontier points from the original sample. Farthest frontier points are chosen using a geometrical approach for estimating extreme points of a class and autoencoder for learning a compressed representation of the data. For experimentation, we have used datasets of MNIST, Iris, credit card fraud detection, Indian Pines, Human Activity Recognition Database.

Keywords: Sample Reduction; Autoencoder; Dimensionality Reduction; Farthest Boundary Point Extraction; Multiclass Classification; SVM; Training Data Reduction

Introduction

Reduction of training data selection from the large dataset using encoder and decoder algorithm without much compromise of accuracy. Classification is one of most salient tasksin Machine learning where a program learns from a given dataset/observation and then classifies new observations into a number of classes or groups. Support vector machine (SVM) is a class of popular learning algorithms that givesfine generalization on classification tasks. The task is to tell whether a given data sample is akin to a particular class ornot.

Two common issues with classification

- **Training time:** It is an important parameter in training any model. However training a SVM model is very time consuming with a large set of samples. Employing coherent algorithms such as dimensionality reduction and training sample reduction the training time can be reduced.
- **Large Dataset**: Another issue is needlessly working with large datasets. In Algorithms like SVM that depend on support vectors i.e. only samples near the decision boundary have an influence on the classification hyper- plane. Finding these samples can greatly reduce time and space complexity for training.

Literature survey

Selecting training points for one-class support vector machines. By Li, Y., *et al.* [7]

Li Proposed this approach in 2011. It selects the peripheral samples among all available training samples. Exterior Points are selected using their k nearest neighbours. If the point is exterior and rests on the concave surface then all of the KNN lies on the one side of the tangent plane. For the concave surfaces it will be true for most of the points.

So a hyperparameter is chosen in the range [0,0.2] as the threshold on the number of nearest neighbours on the particular side of the tangent plane.

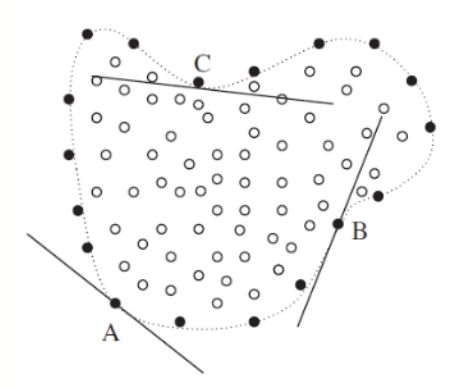


Figure 1: Concave and Convex surfaces [10].

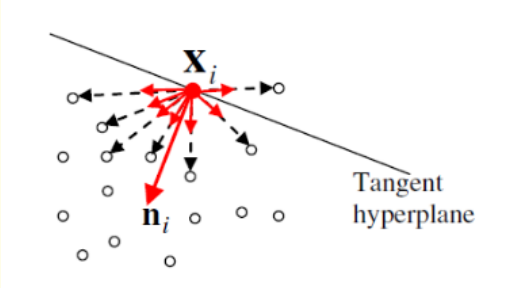


Figure 2: Tangent Hyperplane [10].

This approach uses Nearest Neighbour calculation as the subroutine which increases the computational overhead especially for large or higher dimensional data. Further in practice it is found that on higher dimensional data this method tendsto pick samples with lesser information.

Boundary detection and sample reduction for one-class Support Vector Machines. By Fa Zhun, Ning Ye, Wei Yu, Sheng Xu, Guobao Li

This model was proposed in 2014. This model is effective over high dimensional dataset. It also uses the k nearest neighbour as subroutine. Let xo be the point of consideration and xk be the mean of all of its k nearest neighbors, The hyperplane perpendicular to (xo - xk) and passing through xo will be used to divide.

The neighborhood sphere is divided into two parts D1 and D2. For the exterior point the number of sample points in D1 and D2 will be highly disbalanced.

Distribution property of neighbour and reduction of sample

Here D stands for dataset and x_0 belongs to dataset D.x_i is k-nearest neighbour of $x_0(x_i(i=1,2...,k)\epsilon(x_0))$, $\bar{x_k}$ is mean of the neighbors ($x_k = (1/k)\sum_{i=1}^{k} x_i$).

Neighbourhood of sphere is being divided in 2 parts by hyper-plane AB. Hemishpere in which x_k is located is D1 , and remaining one is D2.

For sample neighbour angle , ϑ is defined as angle between x_0-x_i and x_0 - $\bar{x}_k$. ($\bar{x}_k$ = mean of neighbours ,$x_k = (1/k)\sum_{i=1}^{k} x_i$). So cosine of ϑ is defined as :

$$c_0, i = \cos(\vartheta) = \langle x_0 - \bar{x_k} , x_0 - x_i \rangle / \|x_0 - x_i\| \|x_0 - \bar{x_k}\|$$

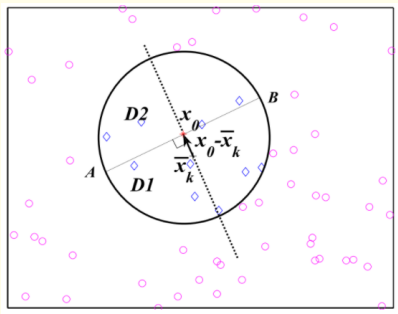


Figure 3: The definitions of the D1 and D2 [9].

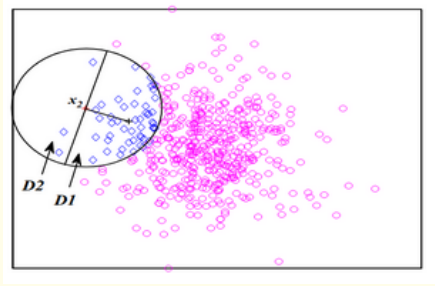


Figure 4: D1 and D2 hemisphere explanation [9].

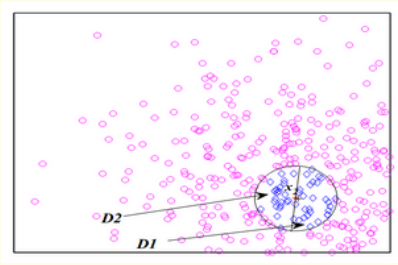


Figure 5: D1 and D2 hemisphere explanation [9].

This approach uses Nearest Neighbour calculation as the subroutine which increases the computational overhead. To overcome that, the authors proposed the parallel computation of KNN in subset. It decreases the complexity of computation from O(n2) to O(nm) where m denotes the average number of samples in the subsets. In practice this method proved to be robust on multidimensional datasets.

Heuristic sample reduction method for support vector data description By Wenzhu Sun, Jianling Qu, Yang Chen, Yazhou Di, Feng Gao

This model uses a heuristic approach to choosing samples that are probable support vectors. This approach uses k- means clustering as a subroutine. The heuristic method can be described in following steps.

Initialize Xtr as normalized training set and empty RedX as reduced set.

- Obtain cluster centers c1, c2, c3... ck by applying kmeans on Xtr
- Calculate di - variance of distance for i^{th} sample to its nearest cluster center.
- Sort all members of Xtr according to di as calculated above
- Add sample having largest di to RedX (reduced set) and note this sample as Xh.
- Remove samples xi from Xtr having inner product set by user.

||Xh, xi|| < threshhold

- Repeat above two steps till Xtr set is empty.

The RedX contains the reduced set that has probable support vectors.

The challenge in this method comes from choosing 'k' value in k-means and setting the distance threshold in for filtering the training set.

Sample reduction using farthest boundary point estimation (FBPE) for Support Vector Data Description (SVDD) By Shamshe Alam, Sanjay Kumar Sonbhadra, Sonali Agarwal, P. Nagabhushan, M. Tanveer

This model was proposed in 2020. This is a geometrical model which assumes the normal distribution of the class samples and the uniform distribution of the outliers. This is a recursive procedure in which the mean of current sets of samples are calculated and then angular observations are performed from the mean point concerning all dimensions. For each observation the point farthest from the mean point is picked into the selected point list and the rest all get discarded.

The point among the selected points which is farthest from the mean point is taken out as the final selected point and rest all points recursively undergo the same procedure till there mean point become zero or all samples are either discarded or moved to finally selected list.

This model performs well on high dimensional dataset. The time complexity of model on high dimensional data (where d is comparable to n) is O(kn2) and on the low dimensional data is O(kdn) here d stands for number of dimension or number of feature, n stands for count of samples and k stands for count of angular observations.

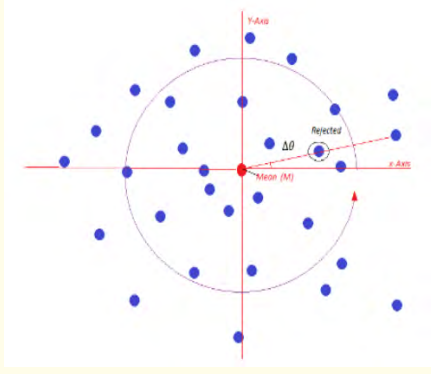


Figure 6: Rejected samples gray; selected sample yellow; selected sample for further processing blue [3].

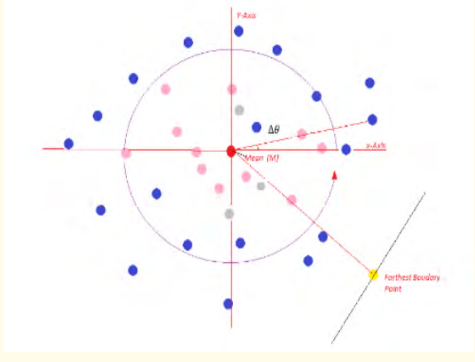


Figure 7: Rejected samples gray; selected sample yellow; selected sample for further processing blue [3].

Weakness

This models fails to extract all the boundary points in case of an open ring like structure where for each point in the interior boundary there exists a point in the exterior boundary which is farther from the mean point. For such a structure the FBPE will reject the interior boundary in the first pass itself.

In figure 7 there are two different classes both of them have the interior and exterior boundary.

Figure 8 is the result of running FBPE on the same dataset.

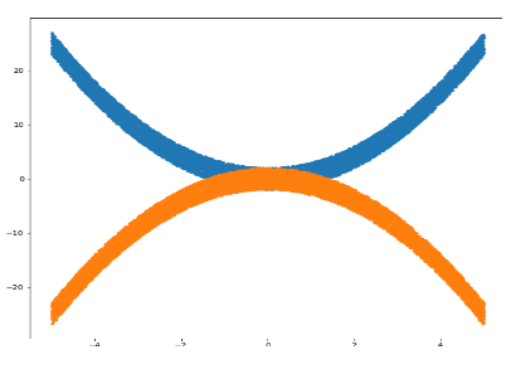


Figure 8: Two class dataset with double boundary.

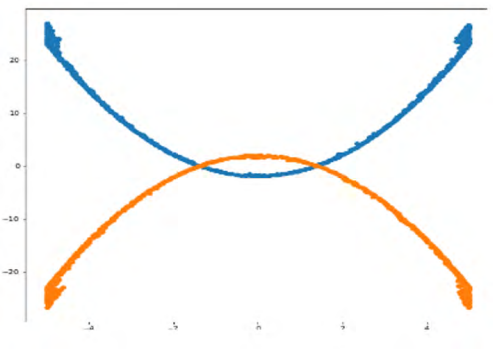


Figure 9: Result After Running FBPE on the dataset.

Auto-encoder based dimensionality reduction By Yasi Wang, Hongxun Yaon, Sicheng Zhao

In this paper authors have explored the ability of Auto-Encoders is responsible for reducing data dimensionality andit is compared with different reduction of dimensionality techniques i.e. LDA, LLE, PCA.

They have conducted their experiments on 2-D and 3-D synthesized datasets to help in proper visualization of dimensionality reduction by all of the dimensionality reduction techniques. Then they also visualise for both MNIST and Olivetti face dataset, the distribution of classes in 2D and 3D spaces and compare those with the ones generated using PCA, LDA, LLE and Isomap. Finally they experimented the effect of the encoding size of Auto-Encoder on the classification accuracy of the softmax layer for both MNIST and Olivetti face dataset.

In the first two experiments they found that all of the methods performed satisfactory but the autoencoder was able to detect the repetitive structures better and in the second experiment Auto-Encoder achieved the best separation between classes. They reasoned the result by the fact that usually high dimensional data lies in intrinsic dimensionality space which in comparison of different dimensionality reduction techniques gives satisfactory result, but the relation between this low dimensional space and original space is very complex in actual real data, therefore this is the reason behind excellence of neural network based methods.

In the last experiment in a non fine-tuned network they found that for simpler datasets like MNIST the accuracy saturates as the encoding size reaches above 10 but in Olivetti face dataset which is much more complex than the previous one the accuracy continues to grow further. They concluded that optimal encoding size is directly proportional to the complexity of the dataset.

Reducing the Dimensionality of a with Neural Networks - G E. Hinton* and R. R. Salakhutdinov

Large dimension is converted to lower dimension using multi layer neural network and that trained encoded representation will be used to reconstruct high dimension. Generally gradient descent are used for correcting or improving the weights but it works only when initial weights are good enough. In this paper, weight initialisation technique has been focused which will allow deep auto-encoder to learn effectively low dimension feature.

Normally, neural network is trained with random weights initially, with objective to minimise reconstruction error. Gradient errors are obtained from chain rule in back propagation. But optimising weights in non linear auto-encoder is difficult task which have multiple hidden layers. In this paper, they have talked about ”pretraining” method for datasets of binary type, which is generalised to real-value dataset.

They have modeled binary data using two layer network. In this stochastic, all binary pixels connected to it. They are binary feature detectors having weights symmetrical in nature. They have defined energy for visible (v) and hidden units (h) on basis of which every possible image is assigned a probability.

$$E(v,h) = -\sum_{i \epsilon pixels} b_i v_i - \sum_{i \epsilon features} b_j h_j - \sum_{i,j} v_i h_j w_{i,j}$$

Where v_i, h_j denotes binary states feature j and pixel i. When binary states are selected for hidden units, "confabulation" is generated and value of vi set as 1 with probability from s logistic function. After that hidden units states are updated, so as weights are updated.

Real valued stochastic nature of hidden units, is unit variance having mean value determined from visible part, which makes low dimensional code uses very well of continuous variable. So this layer by layer training can also be used effectively for other datasets as well for classification problems.

Reducing the number of training samples for fast support vector machine classification by Ravindra Koggalage and Saman Halgamuge Department of Mechanical and Manufacturing Engineering, The University of Melbourne

Support Vector Machine is widely used because of great classification accuracy. It is able to achieve it because of good generalisation skill. But one of the limitations of SVM is size of training dataset. As per SVM methodology construction of hyperplane depends only on some of the training samples not all, only which lies near to hyperplane or decision boundary. In this paper, they have proposed that with using clustering methods, they find initial cluster first and they update it to add more non relevant record, this is being carried out while checking for decision boundary of SVM. In this manner, they attempt to reduce size of samples such that they don't loss accuracy.

While reducing sample size they keep in mind that following parameters doesn't change much i.e. classification accuracy, support vectors and removing those samples which has minimum effect i.e. mainly those samples which are very far from hyper plane or decision boundary. For dataset, they have basically used 2 class dataset, having non-linear separable daa points.

They have used following three steps for sample reduction.

- Initially identifying cluster centers.
- Then identifying available centers which are crisp.
- Finally detecting samples which are required to remove.

Various clustering methods unsupervised or supervised can be used. It's main aim is to first identify initial clusters it is chosen in such manner such that overall complexity of selecting initial cluster should be minimum. Then next step is identifying cluters from records i.e. having a single class which are called crisp cluster. After that sample removal is considered i.e. samples from any particular crisp centre can either be on hyperplane or near to it then those samples of crisp should not be removed, otherwise we remove those sample.

DATASET

We have used these 5 datasets-

MNIST

Dataset of handwritten digits (0 to 9) consisting of 60000 training images and 10000 testing images almost equally distributed between all classes (digits 0 to 9). The dataset is primarily used for training image processing models.

Iris

Iris flower dataset contains different species. The three species are versicolor, virginica, setosa. Each record in dataset has four different features such as width, length of part of flower i.e. petals and sepals. The dataset is mainly used for classification tasks.

Credit Card Fraud Detection dataset

This dataset contains 285000 number of transactions, these transactions are from Europe. Dataset has 2 classes '1' for fraud and '0' otherwise. It has 28 PCA transformed numerical features with actual feature names undisclosed and 2 non transformed features Time and Amount.

Indian pines

It is a dataset consisting of hyperspectral images over Indian Pines in Indiana, US. Here images are formed having 224 spectral bands and of 145x145 pixels. Here 16 labels different classes which are not mutually exclusive. With varying number of samples for eachclass.

Human activity recognition database

This dataset contains measured data on 30 human sub- jects carrying out day-to-day activities with a smart- phone strapped to their waist having multiple inertial sensors. Total of 6 different activities are recorded labelled as laying, standing, walking downstairs, sitting,walking, walking upstairs.

For all rows present in dataset following information can be inferred from it:

- Label of activity it contains.
- From three axes the calculated angular velocityusing gyroscope.
- For human which is performing experiment it'sinteger identifier.
- Total acceleration has three components along different axes, so it contains that acceleration from different axes, it is measured from accelerometer.
- Frequency and time variables 561 size of vectorcontaining different feature.

Proposed methodology

We have experimented with the effect on the accuracy of the ML models by the reduction of the size of training sets of datasets like Indian Pines, Iris, MNIST etc. by methodslike FBPE and Auto-Encoders.

As FBPE is boundary point estimation we primarily target the models that creates boundary between classes for classification but we also used models like CNN and KNN in dataset like mnist and credit card respectively.

We have compared the accuracy of models trained on original dataset with those trained on dataset preprocessedby either of the FBPE or Auto-Encoder. As Auto-Encoder leads to reduction on number of features and FBPE leads tothe reduction of samples we have also trained and compared the accuracy of models trained on dataset preprocessed by both FBPE and Auto-Encoder. As Auto-Encoders performbetter with more samples we first trained autoencoders to get compressed representation of the data then used FBPE for reducing the samples of the resultant dataset.

Following are the brief descriptions of Auto-Encoder and FBPE.

AutoEncoder

Autoencoders normally is build of four parts:

- **Encoder:** This is first part of auto encoder architecture, it is basically responsible for reducing the input dimensions such that eventually we'll have compressed representation of input data.
- **Bottleneck:** This is responsible for containing the compressed representation of data which is given input to the model.
- **Decoder:** This is responsible for reconstructing the encoded data representation such that it tries to closely resemble the input image.
- **Reconstruction Loss:** This is responsible for measuring the performance of the decoder, that is how closely it generates output to the input image.

Farthest boundary point estimation (FBPE) algorithm

Step 1: Initialization

n = S (length of sample)x = Mean of S ϵ Rd

Selected Points = (sample points selected in first round)

Final Selected Points = (Output)

Step 2: Selection of initial samples w.r.t. mean x

For all angles = 0 to 2

For each neighbours in the direction nj

Where j =1,2,3 ⋯ n

Select the farthest point in np direction

and add in Selected Points, reject other points.

Step 3: Selection of final points from samples

Remove the farthest neighbor from the selected points and store in final selected Points as boundary point and store in Final Selected Points.

Step 4: Take mean of the selected points

X = Mean of selected points.

if x, 0 Go to step 2.

Step 5: Return final selected points.

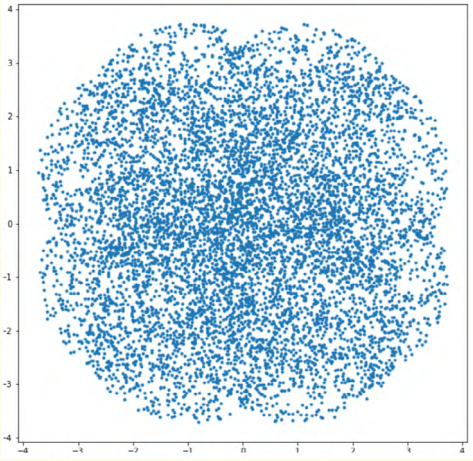


Figure 10: A data point cluster with both concave and convex surfaces.

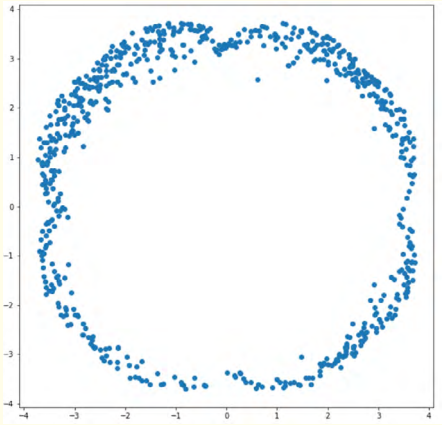


Figure 11: Result After Running FBPE on the cluster.

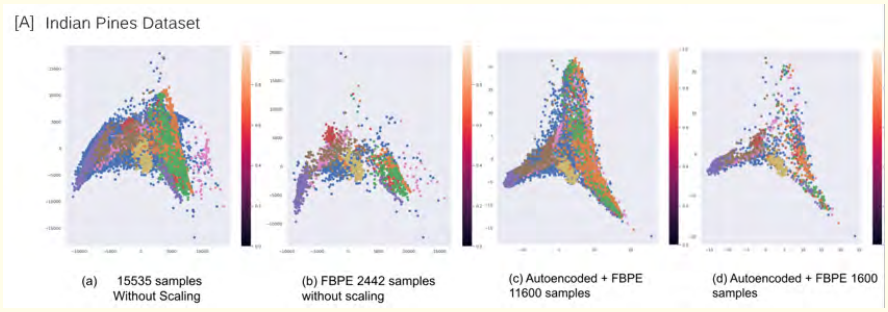
Experimental results

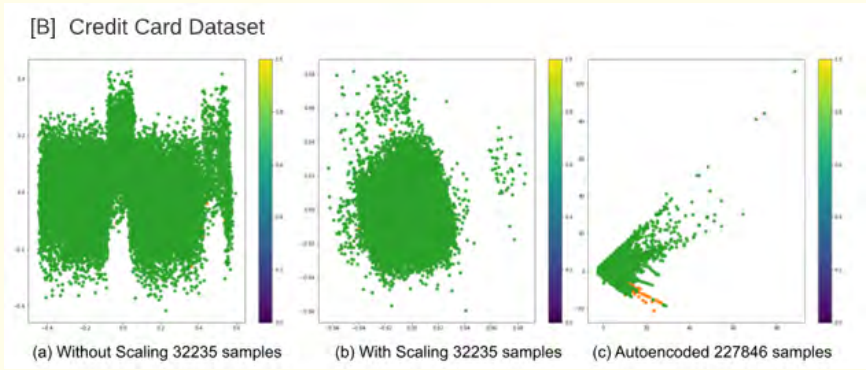
Dataset	Model	Initial number of training samples	Initial number of features	Final number of training samples	Final number of features	Percentage reduction	Accuracy
Indian Pines	Autoencoder + FBPE + SVM Linear Kernel	14025	200	3670	50	74.84%	93.64%
Indian Pines	Autoencoder + FBPE + SVM RBF Kernel	14025	200	3670	50	74.84%	79.32%
Indian Pines	Autoencoder + FBPE + SVM Linear Kernel	14025	200	1599	100	94.3%	95%
Indian Pines	Autoencoder + FBPE + SVM rbf Kernel	14025	200	1599	100	94.3%	77.02%
Indian Pines	Autoencoder + FBPE + SVM Linear Kernel	14025	200	11652	100	58.46%	99.1%
Indian Pines	Autoencoder + FBPE + SVM rbf Kernel	14025	200	11652	100	58.46%	88.2%
Indian Pines	SVM Linear without Scaling	15549	200	15549	200	0%	91.25%
Indian Pines	SVM rbf without Scaling	15549	200	15549	200	0%	61.02%
Indian Pines	FBPE + SVM Linear without Scaling	16820	200	2442	200	85.48%	48.75%
Indian Pines	FBPE + SVM rbf without Scaling	16820	200	2442	200	85.48%	29.25%
Indian Pines	FBPE + SVM Linear without Scaling	16820	200	15375	200	8.59%	76.74%
Indian Pines	FBPE + SVM rbf without Scaling	16820	200	15375	200	8.59%	60.97%

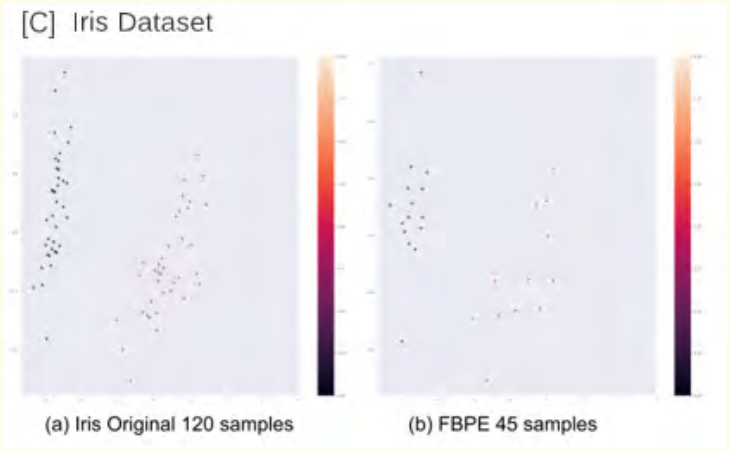
Indian Pines	FBPE + SVM Linear with Scaling	16820	200	4960	200	70.51%	36.79%
Indian Pines	FBPE + SVM rbf with Scaling	16820	200	4960	200	70.51%	31.67%
Indian Pines	FBPE + SVM Linear with Scaling	16820	200	11226	200	33.26%	65.58%
Indian Pines	FBPE + SVM rbf with Scaling	16820	200	11226	200	33.26%	60.83%
Human Activity	Autoencoder + FBPE + SVM linear Kernel	7299	561	1900	140	93.5%	96.5%
Human Activity	Autoencoder + FBPE + SVM rbf Kernel	7299	561	1900	140	93.5%	95.56%
Human Activity	Autoencoder + FBPE + SVM Linear Kernel	7299	561	4347	140	85.14%	97%
Human Activity	Autoencoder + FBPE + SVM rbf Kernel	7299	561	4347	140	85.14%	97.16%
Human Activity	SVM Linear	7352	561	7352	561	0%	96.4%
Human Activity	SVM rbf	7352	561	7352	561	0%	95.01%
Human Activity	FBPE + SVM linear	7299	561	1059	561	85.49%	89.95%
Human Activity	FBPE + SVM rbf	7299	561	1059	561	85.49%	86.9%
Human Activity	FBPE + SVM linear	7299	561	2094	561	71.31%	92.43%
Human Activity	FBPE + SVM rbf	7299	561	2094	561	71.31%	90.83%
Credit Card	FBPE + SVM Linear Kernel + Without Scaling	227846	29	32235	29	85.85%	99.81%
Credit Card	FBPE + SVM RBF Kernel + Without Scaling	227846	29	32235	29	85.85%	99.81%
Credit Card	FBPE+SVM Linear Kernel + With Scaling	227846	29	32235	29	85.85%	99.91%
Credit Card	FBPE+SVM RBF Kernel + With Scaling	227846	29	32235	29	85.85%	99.91%
Credit Card	FBPE + KNN + With Scaling	227846	29	32235	29	85.85%	78.28%
Credit Card	KNN + With scaling	227846	29	227459	29	0.17%	89.5%
Credit Card	Autoencoder + KNN + With Scaling	227845	29	227845	3	89.66%	50.36%
Credit Card	Autoencoder + SVM+ With Scaling	227845	29	227845	3	89.66%	99.82%
Iris	FBPE + SVM Linear kernel	150	5	58	5	61.33%	100%
Iris	FBPE + SVM RBF kernel	150	5	58	5	61.33%	91.6%
Iris	SVM Linear Kernel	150	5	150	5	0%	100%
Iris	SVM RBF Kernel	150	5	150	5	0%	100%
MNIST	CNN	60000	784	60000	784	0%	98.7%
MNIST	FBPE + CNN	60000	784	25781	784	57.03%	94.95%
MNIST	SVM	60000	784	60000	784	0%	98%

MNIST	FBPE + SVM	60000	784	25781	784	57.03%	91%
MNIST	Autoencoder + SVM	60000	784	60000	128	83.67%	96%
MNIST	Autoencoder + FBPE + SVM	60000	784	38021	128	89.65%	94%
MNIST	Autoencoder(768 to 3) + SVM	60000	784	60000	3	99.62%	77%
MNIST	Autoencoder (768 to 3) + FBPE + SVM	60000	784	20099	3	99.87%	80%

Table 1: Experimental Summary.







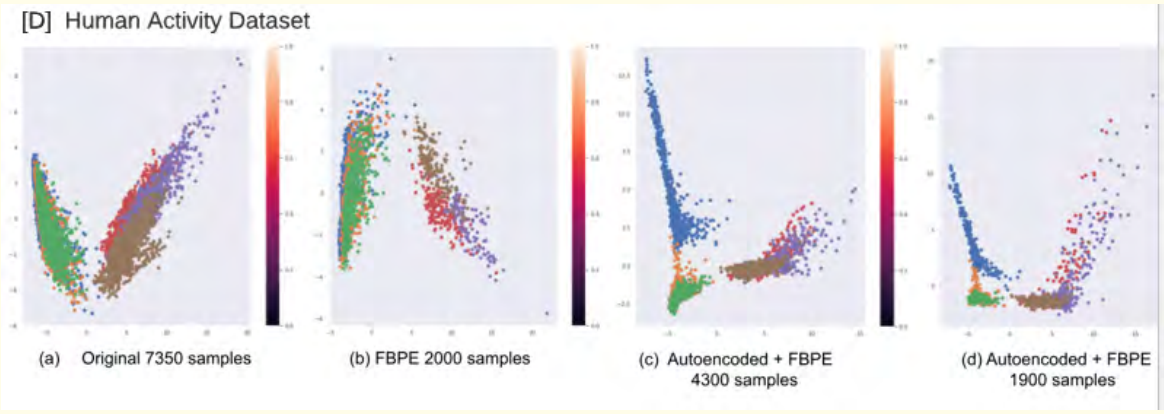


Figure a: Visualizing sample reduction using PCA with principal components = 2 on reduced datasets.

Indian pines

The dataset contains image in 145 x 145 pixels in 200 multispectral bands. We first preprocess the dataset, extracting pixels and storing along with corresponding ground truth label as class label. This yields a labelled dataset containing 21025 samples with 200 features (spectral reflectance bands). The dataset is now split 80: 20 between train and test segments. After this various models in permutations of Autoencoder, FBPE and SVM are implemented. Since the pixel values are integer values, so all models that do not involve autoencoder have variants with and without scaling.

Plain SVM without scaling the 200 features of the dataset, training on 15549 samples gives 91.25% testaccuracy with linear kernel and 61.02% test accuracywith RBF kernel.

FBPE + SVM

On unscaled data

The 16820 training samples are passed through FBPE sample reduction for various classSizes yielding reduction to 2442 and 15375 samples. A linear and RBF kernel SVM are trained on this reduced data separately and tested on the 20% split data of 4205 samples.

Applying FBPE and reducing this dataset of 16820 to 2442 training samples yields 48.75% and 29.25% test accuracy on linear and rbf kernels respectively.

While applying FBPE and reducing this dataset of 16820 to 15375 training samples yields 76.74% and 60.97%test accuracy on linear and rbf kernels respectively.

On scaled data

Data is MinMax Scaled for every spectral band individually with 80: 20:: train: test split. The 16820 training samples are passed through FBPE samplereduction yielding 11226 samples and 4960 for different runs. A linear and RBF kernel SVM are trained on this reduced data separately and tested on the 20% split data.

Autoencoder

The entire dataset is Min-Max scaled then for autoencoding the entire dataset of 21025 samples is passed through an autoencoder. The first layer is a Dense layer of twice the input size followed by Batch Normalization and Leaky relu. Subsequent two layers have similar structure with the number of units halving each from the previous layer.

The encoded version contains two reductions - 100 features reduced from 200 50 features reduced from 200. The encoded dataset is split 14025: 7000 for train: test set. For interest of experiment an unscaled version of autoencoder is also trained but it fails to converge as evident from monitoring training accuracy. This happens because of integer overflow encountered as a consequence of working on unscaled data.

Autoencoder + FBPE + SVM

Using Autoencoder reducing 200 features to 100 features, appling FBPE to reduce to 1599 training samples gives 95% test

accuracy on Linear kernel SVM and 77.02% accuracy on RBF kernel SVM. Similarly applying FBPE to reduce to 11652 training samples gives 99.1% testaccuracy on linear kernel and 88.20% test accuracy on RBF kernel SVM. Using Autoencoder reducing 200 features to 50 features, appling FBPE to reduce to 3670 training samples gives 93.64% test accuracy on Linear kernel SVM and 79.32% accuracy on RBF kernel SVM.

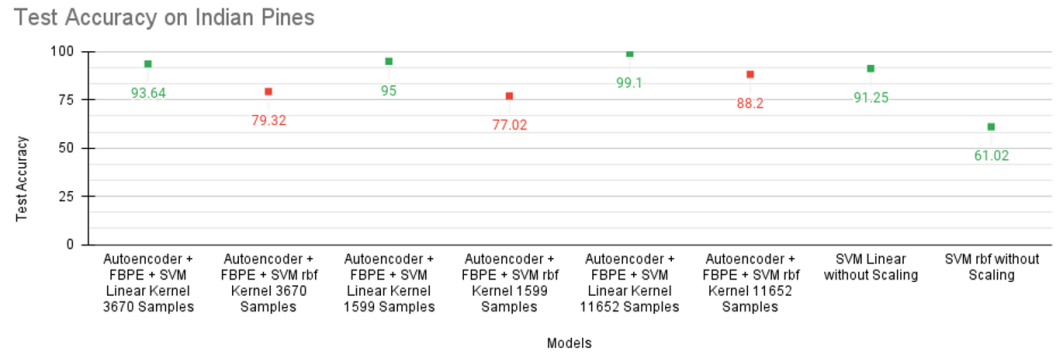


Figure 12: Test accuracy vs model for Indian Pines dataset.

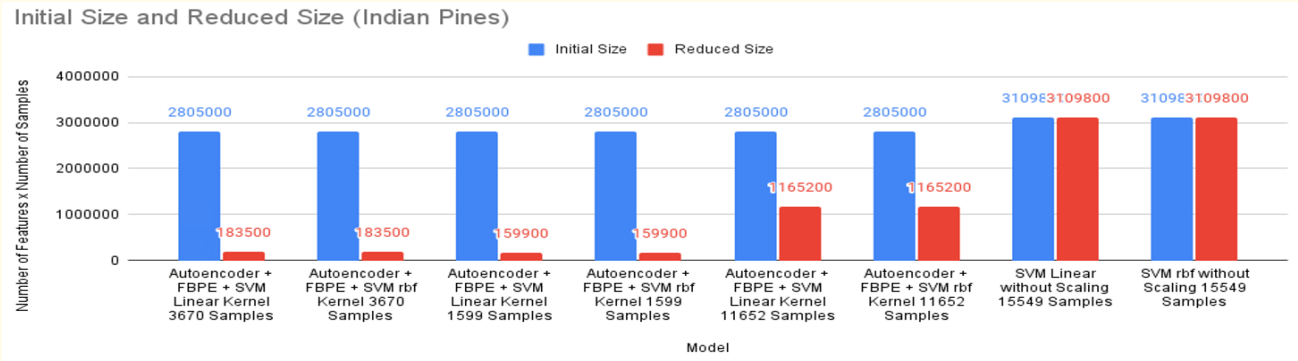


Figure 13: Initial Size and Reduced Size (Indian Pines).

Human activity recognition

The dataset contains 10000 samples with 563 features having 561 features pertaining to measurements frominertial sensors, a feature on subject identifier amongthe 30 participants and class label on activity being performed. The column containing subject identifier is dropped yielding a 561 feature dataset excluding the class label. The dataset is now split 70: 30 betweentrain and test segments. After this various models in permutations of Autoencoder, FBPE and SVM are implemented.

Plain SVM

The 70: 30:: train: test split is used to train and test a linear and rbf kernel SVM separately. SVM with linear kernel trained on 70% of original dataset yields 96.40% test accuracy and SVM with RBF Kernel yields 95.01% accuracy.

Autoencoder

The entire dataset is Min-Max scaled then for autoencoding the entire dataset of 10229 samples is passed through an autoencoder. The first layer is a Dense layer of twice the input size followed by Batch Normalization and Leaky relu. Subsequent three layers have similar structure with the number of units halving each from the previous layer. The encoded version contains 140 features reduced from 561. The encoded dataset is split 7299: 3000 for train: test set.

Autoencoder + FBPE + SVM

After trying FBPE sample reduction on training set with various classSizes a reduction to 1900 samples and another run to 4347 samples is achieved which isthen used to train a SVM with Linear and RBF Kernel separately. Autoencoder reduced set with 140 features then FBPE reduced to 1900 samples on Linear kernel SVM gives 96.5% and RBF Kernel SVM gives 95.56% test accuracy while 140 features set then FBPE reducedto 4347 samples yields 97% test accuracy with Linear kernel SVM and 97.16% test accuracy with RBF Kernel SVM.

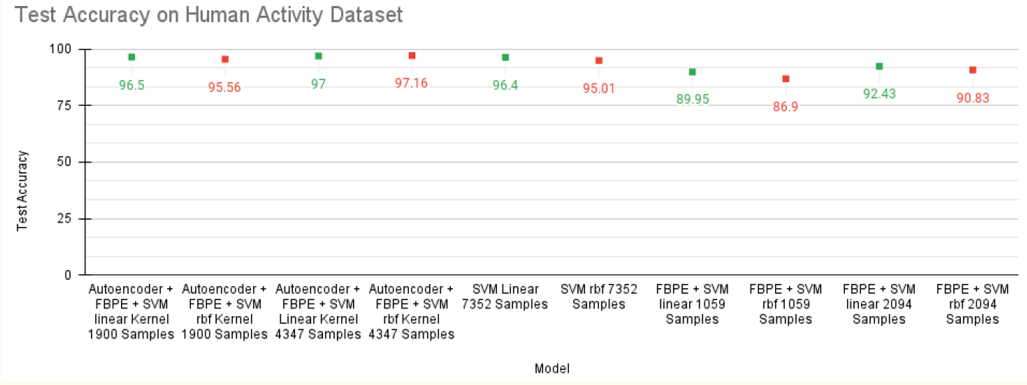


Figure 14: Test accuracy VS Model for Human Activity dataset.

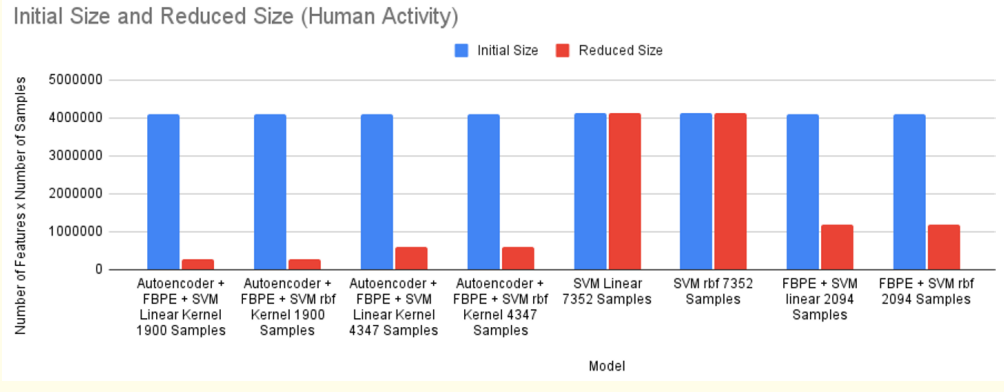


Figure 15: Initial Size and Reduced Size (Human Activity).

Credit card fraud detection

It contains transaction datas of European credit card holders. It contains transactions of 2 days, having fraud and non fraud transactions. Total transaction is 284,807 out of which 492 cases are fraud. From above we can see that the dataset is unbalanced, i.e. fraud cases constitute only 0.172% of total transactions. All attribute values are numeric in nature. As per kaggle, to ensure confidentiality of dataset attributes, original features are not provided. They have used PCA to obtain 28 features such as V1,V2,.. v28. Remaining two features which are left is amount and time. As per kaggle, time attribute is responsible for accounting difference between first transaction in the list and from other transactions respectively. Class attribute contains 0 and 1 which means 0 for non fraud cases and 1 for fraud cases.

As stated above, we have tried different combinations of FBPE, Autoencoder, Classification algorithms. Number of training samples initially were 227846. After applying FBPE on the training dataset we have reduced it to 32235 numbers of training samples. We have droppedthe time label in our dataset in some cases which makes the number of features to be used is 29.

FBPE + SVM

- **Scaled Dataset:** Here we have scaled all columns as per min-max scaling, We have used two different types of kernel for SVM. For FBPE with linear kernel 99.91% accuracy is obtained and with rbf kernel accuracy is 99.91%.
- **Unscaled Dataset:** We have not scaled the amount attribute. We have used two different types of kernel for SVM. For FBPE with linear kernel accuracy obtained is 99.81%. and with rbf kernel accuracy is 99.81%.

FBPE + KNN

- **Scaled Dataset:** Here we have scaled all columns as per min-max scaling, After applying KNN model on FBPE reduced dataset, we have obtained accuracy of 78.28%.
- **KNN:** After applying KNN on a complete scaled dataset, we have obtained accuracy of 89.5%.

Autoencoder + SVM

- **Scaled Dataset:** We have scaled dataset on amount attribute with min-max scaling, and reduced the number of features to 3. After applying Autoencoder on training dataset, and then we have applied SVM model. We have obtained accuracy of 99.82%.

Autoencoder + KNN

- **Scaled Dataset:** We have scaled dataset on amount attribute with min-max scaling, and reduced the number of features to 3. After applying Autoencoder on training dataset, and then we have applied KNN model. We have obtained accuracy of 50.36%.

IRIS

It is iris species dataset which include three iris species. Total 50 samples of each species are taken in dataset. Total size of dataset in this case is 150 with properties of flowers as attributes. As per kaggle dataset, one of the flower species is separable from remaining 2 linearly, while remaining two are not separable linearly from each other.

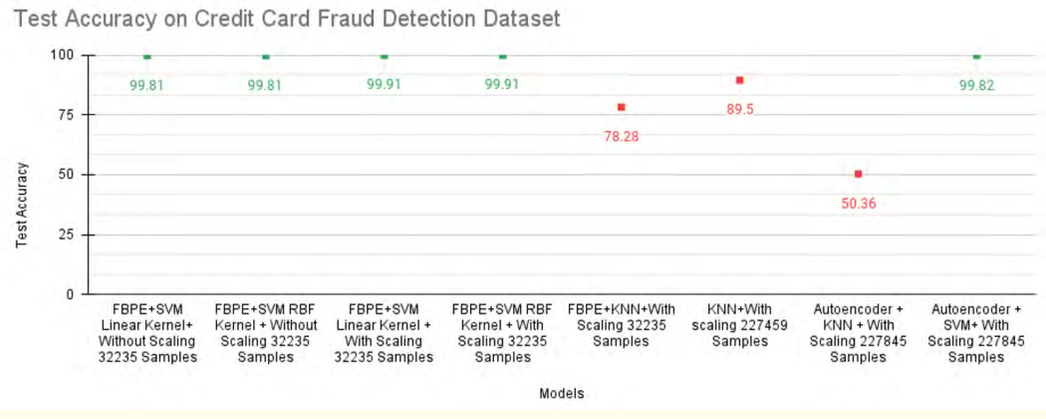


Figure 16: Test accuracy VS Model for CreditCard Fraud Detection Dataset.

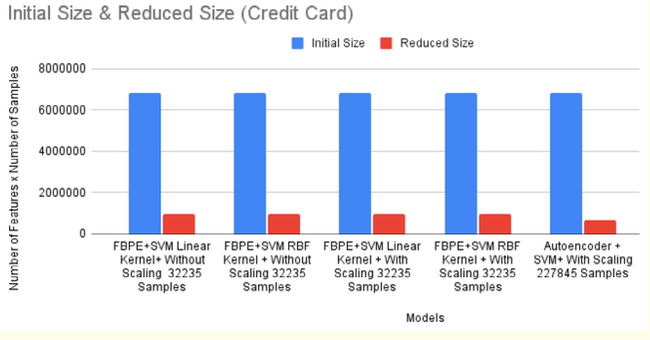


Figure 17: Initial Size and Reduced Size (Credit Card).

Following are the attributes of iris species, Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm Species. Since the number of features are 4, there is no requirement to run an autoencoder on this dataset. So we have applied only FBPE with different classification algorithms on it. After applying Fbpe algorithm on iris dataset, size of dataset is reduced from 150 to 58.

FBPE + SVM

We have used different kernels for SVM. For SVM with linear kernel, accuracy obtained is 100%, whereas with rbf kernel accuracy obtained is 91.6%.

SVM

In this case, we have applied SVM with different kernels on a complete dataset. So with linear kernel, on 150 samples, we have obtained accuracy of 100%. Again with rbf kernel we have obtained accuracy of 100%.

MNIST

CNN with 5 million parameters with one convolutional and one dense layer gives accuracy of 98.7% on the original data set and gives 94.95% accuracy when the training samples were reduced to 25K from 60K using FBPE.

While using SVM with RBF kernel the model train on the original dataset with 60K samples gave 98% accuracy whereas on the dataset reduced to 25K using FBPE the model yields 91% accuracy.

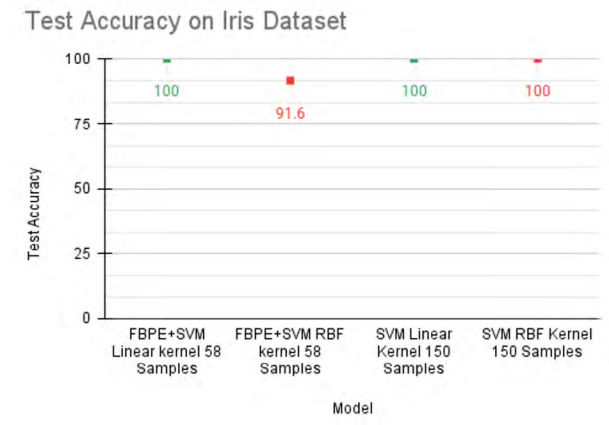


Figure 18: Test accuracy VS Model for Iris dataset.

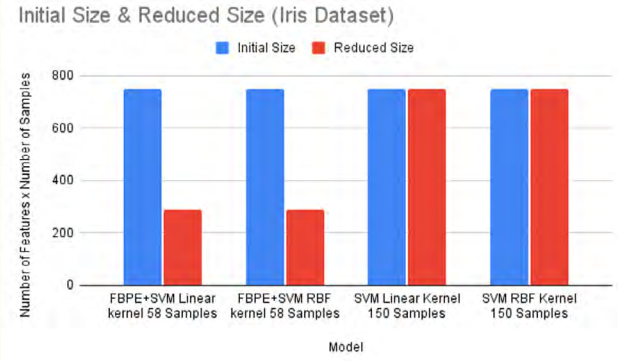


Figure 19: Initial Size and Reduced Size (Iris Dataset).

We also used Auto-Encoders in which number of features have been reduced from 784 to 128 and 3 and then compared the accuracies yielded by SVM models with RBF kernel trained with and without sample reduction. For the dataset with 128 features and 60K samples the SVM gives 96% accuracy whereas for a dataset of 38K samples obtained using FBPE the SVM gives 94% accuracy.

In the case of 3 features with 60k samples SVM yields 77% but with 20K samples it was 80%.

Conclusion

We have successfully applied combinations of fbpe algorithm, auto-encoder and classification algorithms on various datasets. For MNIST dataset, number of samples have been reduced from 60000 to 25781 and number of features from 784 to 128 and then to 3 with best accuracy of 96% from all combination of models using FBPE. For Iris number of samples have been reduced from 150 to 58 with 4 features with best accuracy of 100% from all combination of models using FBPE. For Credit number of samples have been reduced from 227846 to 32235 and number of features from 29 to 3 with best accuracy of 99.91% from all combination of models using FBPE. For Human Activity number of samples have been reduced from 7299 to 1059 and number of features have been reduced from 561 to 140 with best accuracy of 97.16% from all combination of models using FBPE. For India Pines dataset, we have reduced number of samples have been reduced from 14025 to 1599 and number of features have been reduced from 200 to 100 with best accuracy of 99.1% from all combination of models using FBPE. We have used classification algorithms like CNN, SVM,

KNN as per the dataset. So we have successfully reduced number of samples and number of features while preserving accuracy of classification models.

Dataset	Y. Li (Li, 2011)	Fa Zhu (Zhu., *et al.* 2014)	FBPE (2020) S Alam., *et al.*	FBPE + Auto-Encoder
MNIST	91.36%	92.06%	91.89%	94%
Indian Pines	82.36%	78.27%	87.16%	99.1%

Table 2: Comparative Results of Average Accuracy.

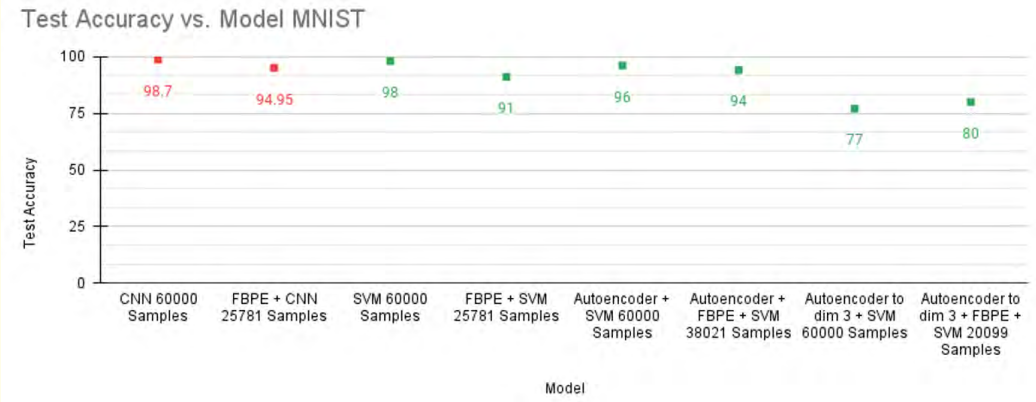


Figure 20: Test accuracy vs model for MNIST dataset.

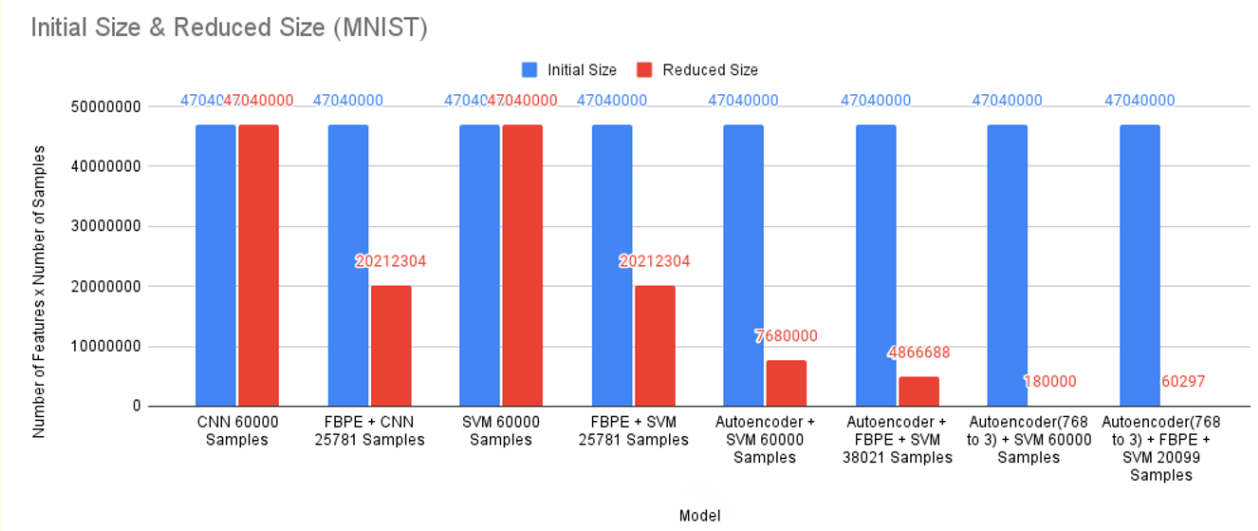


Figure 21: Initial Size and Reduced Size (MNIST).

Bibliography

1. Bagesh Kumar., *et al*. "A fast learning algorithm for One-Class Slab Support Vector Machines". Indian Institute of Information Technology, Allahabad, India.

2. Aha DW., *et al*. "Instance-based learning algorithms". *Machine Learning* 6 (1991): 37-66.

3. Alam Shamshe., *et al*. "Sample reduction using farthest boundary point estimation (FBPE) for support vector data description (SVDD)". *Pattern Recognition Letters* 131 (2020): 268-276.

4. Angiulli F. "Prototype-based domain description for one-class classification". *IEEE Transactions on Pattern Analysis* 34 (2012): 1131-1144.

5. Yasi Wang., et al. "Auto-encoder based dimensionality reduction". School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China.

6. Chalapathy R and Chawla S. "Deep learning for anomaly detection: A survey". arXiv preprint arXiv:1901.03407 (2019).

7. Domingues R., *et al*. "A comparative evaluation of outlier detection algorithms: Experiments and analyses". *Pattern Recognition* 74 (2018): 406-421.

8. Fan Cheng., *et al*. "A subregion division based multi-objective evolutionary algorithm for SVM training set selection". *Neurocomputing* 394 (2020): 70-83.

9. Gates GW. "The reduced nearest neighbor rule". *IEEE Transactions on Information Theory* 18 (1972): 431-433.

10. Wei Wang., *et al*. "Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction". Center for Research on Intelligent Perception and Computing, CRIPAC, Nat'l Lab of Pattern Recognition, Institute of Automation Chinese Academy of Sciences Nat'l Eng. Lab for Video Technology, Key Lab. of Machine Perception (MoE), Sch'l of EECS, Peking University, Beijing, China.

11. Hart PE. "The condensed nearest neighbor rule". *IEEE Transactions on Information Theory* 14 (1968): 515-516.

12. Hastie T and Tibshirani R. "Discriminant adaptive nearest neighbor classification". *IEEE Transactions on Pattern Analysis* 18 (1996): 607-616.

13. Wenzhu SUN., *et al*. "Heuristic sample reduction method for support vector data description". Naval Aeronautical Engineering Institute, Qingdao Branch, Qingdao, P.R. China.

14. Ji M and Xing HJ. "Adaptive-weighted one-class support vector machine for outlier detection". in: Control And Decision Conference (CCDC), 2017 29th Chinese, IEEE (2017): 1766-1771.

15. Latorre Javier., *et al*. "Effect of data reduction on sequence-to-sequence neural tts". ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, (2019).

16. Li Y. "Selecting training points for one-class support vector machines". *Pattern Recognition Letters* 32 (2011): 1517-1522.

17. Ougiaroglou Stefanos., *et al*. "Exploring the effect of data reduction on Neural Network and Support Vector Machine classification". *Neurocomputing* 280 (2018): 101-110.

18. GE Hinton and RR Salakhutdinov. "Reducing the Dimensionality of a with Neural Networks".

19. Reducing the Number of Training Samples for Fast Support Vector Machine Classification by Ravindra Koggalage and Saman Halgamuge Department of Mechanical and Manufacturing Engineering, The University of Melbourne.

20. Rico-Juan JR and I nesta JM. "New rank methods for reducing the size of the training set using the nearest neighbor rule". *Pattern Recognition Letters* 33 (2012): 654-660.

21. Ritter G., *et al*. "An algorithm for a selective nearest neighbor decision rule". *IEEE T Inform Theory* 21 (1975): 665-669.

22. Venelin Valkov. "Credit Card Fraud Detection using Autoencoders in Keras" (2017).

23. Wilson DR and Martinez TR. "Reduction techniques for instance-based learning algorithms". *Machine Learning* 38 (2000): 257-286.

24. Zhu, F., *et al*. "Boundary detection and sample reduction for one-class support vector machines". *Neurocomputing* 123 (2014): 166-173.

25. Zvarevashe Kudakwashe Olugbara Oludayo. "Ensemble Learning of Hybrid Acoustic Features for Speech Emotion Recognition". *Algorithms* (2020).