Perfecting Counterfeit Banknote Detection-A Classification Strategy

Malladi Tejasvi, A.Nayeemulla Khan, A.Shahina

***Abstract*: Machine learning algorithms for classification use the data provided to learn a function that would discriminate between the classes. It is this learning that dictates how well a classifier is able to approximate the function, when presented with unseen data. Counterfeit banknotes are a scourge to every nation. Automated processes to quickly detect counterfeit notes with high accuracy are the essential need. We employ various classification and dimensionality reduction techniques to achieve perfect classification of the Banknote authentication dataset [1] using Artificial Neural Network, Support Vector Machine and K Nearest Neighbours classifiers.**

***Index Terms*: Machine Learning, Neural Networks, counterfeit Banknote detection, Supervised Learning.**

I. INTRODUCTION

The task of learning and positioning decision boundaries separating classes for a given data is a classification problem in machine learning. The decision boundary is a curve that separates data points which fall into one category of data from another. A single decision boundary can split the input space into two parts, and hence is capable of classifying data into two categories. Multiple decision boundaries create multiple regions, formed by their intersections of which each region can represent a class of data. We use this idea to address the banknote classification task. The problem of identifying a currency note as fake or real is a classification problem, which aims at automatically separating the fake notes from the genuine ones and is addressed here. The Machine learning classifiers considered here include perceptron model, decision tree classifier, logistic regression classifier, artificial neural network classifier or multi layer perceptron, support vector machine and K Nearest Neighbours (KNN). Training the classifiers using a training set enables the underlying algorithm to closely approximate a mathematical function that would have generated the labels of the training data. This helps the classifiers to classify a test data point for which the class label is to be found (the solution of our problem statement).

High speed automated counterfeit banknote detection with high accuracy is the need of the hour for most of the institutions right from central banks of countries to small mom and pop stores. With criminals trying to defraud gullible customers or insurgents from hostile countries trying to destabilize the economy of a target country, counterfeit notes may be employed as tools for the same.

Revised Manuscript Received on April 07, 2019.

Malladi Tejasvi, School of Computing Sciences and Engineering, VIT Chennai, Tamil Nadu India.

A.Nayeemulla Khan, School of Computing Sciences and Engineering, VIT Chennai, Tamil Nadu India.

A.Shahina, Department Of Information Technology, SSN College of Engineering, Chennai, Tamil Nadu,

Considering the seriousness of the issue, previous work has focused on detecting specific aspects of banknotes. This was used to detect portraits in the banknotes using CNN. This assisted in authentication [2]. Multispectral images in the visual and infrared spectrum are used to derive features from Korean banknotes. In reference [3] a Gaussian ML classifier as Fischer linear discriminant were used to achieve perfect classification. However the issue is that it needs multispectral images. In a similar work, SVM based classifier that works on security features extracted from Bangladesh banknotes performed with high accuracy. An ordinary mobile camera was used for image capture. The disadvantage with this system is the lack of speed/automation [4]. We use the banknote authentication dataset that has been procured using an industrial camera. The whole image is used for processing and no specific features/objects are extracted. We show simple features derived from wavelet transform achieve high accuracy.

II. DATABASE – BANKNOTE AUTHENTICAION DATASET

The banknote authentication dataset [1] comprises of features of 1372 greyscale bank note images each of size 400x400 pixels, which were captured using an industrial camera. The features were extracted from the images after subjecting them to wavelet transform and the extracted features are Variance (VAR), Skewness (SKE), Kurtosis (KUR) and Entropy (ENT). The dataset consists of just these four features and class labels. The dataset was conceived to distinguish between counterfeit and genuine banknotes. Of the 1372 instances, 610 are encoded as 1 denoting genuine banknotes, and the rest are encoded as 0.The statistical parameters of the data are shown in Table 1.

Table 1. Table showing the statistical parameters of features of banknote dataset. Where SD denotes Standard Deviation and % denotes percentile.

	Variance	Skewness	Kurtosis	Entropy
Count	1372	1372	1372	1372
Mean	0.433	1.922	1.398	-1.191
SD	2.842	5.869	4.31	2.101
Min	-7.042	-13.773	-5.286	-8.542
25%	-1.733	-1.708	-1.575	-2.413
50%	0.496	2.319	0.617	-0.5866
75%	2.821	6.814	3.179	0.395
Max	6.824	12.951	17.927	2.449

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

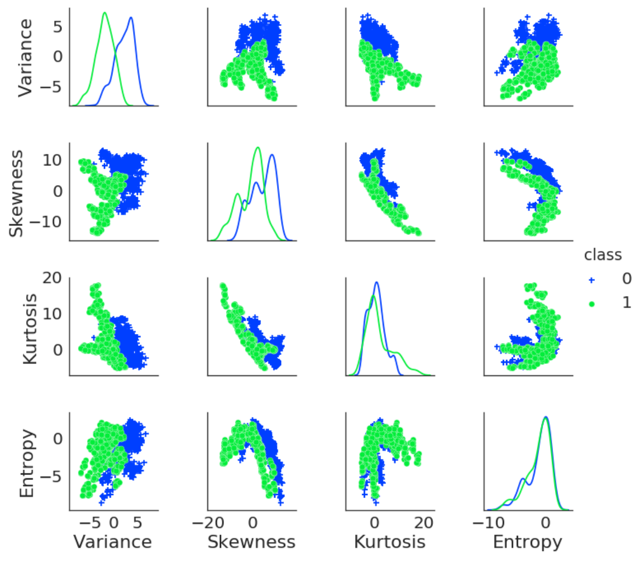

Fig. 1. A pair-plot between all the pairs of attributes in the banknote dataset.

From the pair plots of the attributes is shown in Fig. 1, we infer the following:

1. The variance of both the classes is approximately normally distributed.
2. Both the classes appear right skewed.
3. The Kurtosis of class 0 is approximately normally distributed and that of class 1 is right skewed with high value outliers.
4. The entropy of both the classes happens to follow a left-skewed distribution, indicating the existence of low valued outliers.

In all the experiments conducted, 20% of the examples in the dataset were used for testing and remaining 80% were used for training the model.

III. EXPERIMENTS

The work flow of the experiments carried out would be as shown in Fig. 2. The first step would be data collection, where the appropriate dataset is identified and procured. Then the dataset is pre-processed, this step would include exploratory data analysis. Now, the data is split in 80:20 ratio as training and testing set respectively. Then an appropriate algorithm is chosen, trained and hyper parameters are tuned to give the best model. Then the model would be tested on the test data and if the accuracy is satisfactory, we can declare the model as a accurate otherwise we move on to find another classifier that is suitable.

Exploratory data analysis was performed on the bank note authentication dataset. The scatter plot for all combinations of features in the dataset is shown in Fig. 1. A heat map denoting the correlation between the features of banknotes is shown in Fig. 3.

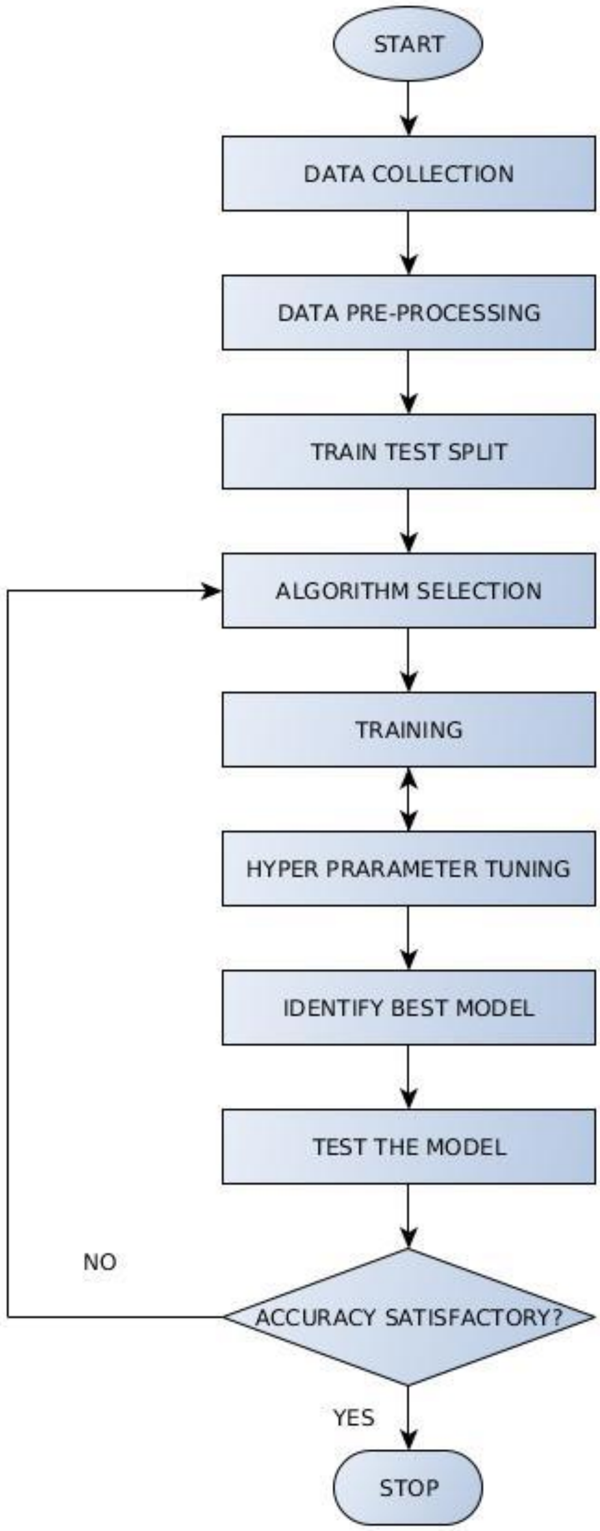

Fig. 2. The work flow of experiments conducted.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

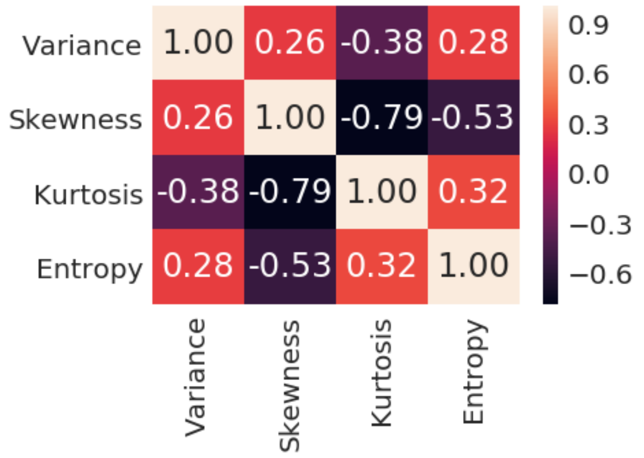

Fig. 3. A heat map showing the correlation between the features in the banknote dataset.

An elbow graph, which is usually plotted to find the 'k' in kmeans clustering, is plotted in Fig. 4.

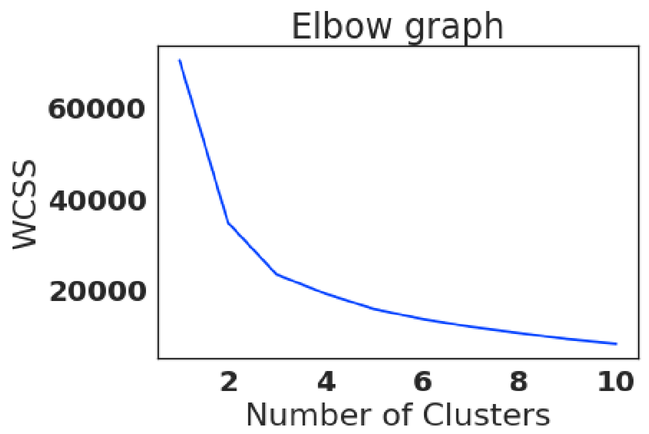

Fig. 4. An elbow graph to find the number clusters required to represent the data.

Principle Component Analysis (PCA) was applied on the dataset and the variance of each feature of the banknote dataset is shown in Table 2.

Table 2. Table showing the variance of each feature of the banknote dataset obtained from PCA

Feature	**Variance**
Variance	0.548
Skewness	0.320
Kurtosis	0.089
Entropy	0.043

From the exploratory analysis we infer the following:

1. The pair plot in Fig. 1 reveals that the data is **not linearly separable** in 2D, hence suitable choices need to be made when choosing the classifier.
2. From the heat map shown in Fig 3, it was found that there was no significant correlation between the features in the dataset, except there is a correlation between skewness and kurtosis.
3. From the graph shown in Fig 4, it could be noted that there is a fall in the within cluster sum of squares(WCSS) error(1), where N denotes number of training examples, k is the number of clusters and $\mu^{(j)}$ is the centroid of the j^{th} cluster. In the figure, there is a sharp fall in the value of error, when number of clusters is between 2 and 3, indicating that there are approximately two or three clusters in the data. This is consistent with the fact that there are two classes in the banknote dataset.
4. From the results of PCA analysis, shown in Table 2 it was observed that the two features namely variance and skewness in the banknote dataset capture most of the variance, indicating that potential information to be learnt is present in these two parameters. This could mean that it may be sufficient to train the classifiers with just these two parameters. This finding may be significant because this can mean a reduction in computational cost and also reduction in the cost of computing the other two parameters from the wavelet transform.
   This could mean that it may be sufficient to train the classifiers with just these two parameters. This finding may be significant because this can mean a reduction in computational cost and also reduction in the cost of computing the other two parameters from the wavelet transform.

$$WCSS = \sum_{i=1}^{N}\sum_{j=1}^{k} w^{(i,j)} \left\| x^{(i)} - \mu^{(j)} \right\|^2 \quad (1)$$

One of the interesting observations was that the feature importance graph(Fig. 5) plotted based on the learning of decision tree classifier supported the observation of PCA that the features, variance and skewness held most useful information in the data.

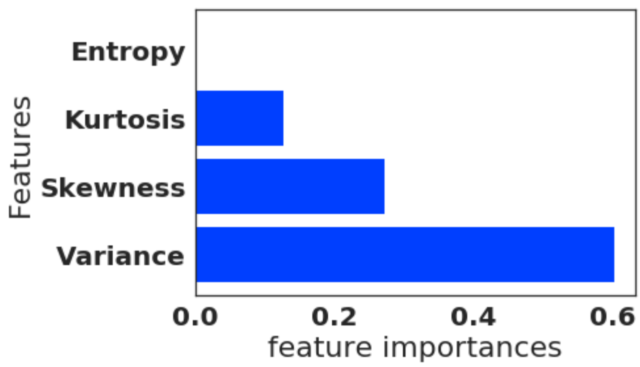

Fig. 5. Feature Importance graph from decision tree classifier.

After observing that the features skewness and variance held most of the information in the data, the aforementioned classifiers were also trained and tested by using only the features, variance and skewness.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

IV. MODELING APPROACHES

A. Overview

We consider the following six different classification algorithms for the task. They are briefly discussed in this section- Perceptron, Decision Tree, Logistic Regression, Artificial Neural Network, Support Vector Machine and K Nearest Neighbour. All the four attributes were used for training the classifiers.

B. Perceptron Learning Model

The simplest of learning algorithms is the perceptron learning algorithm. The performance of this algorithm and the decision surface created will help understand the improvements/models to use to achieve improved performance.

The perceptron learning algorithm functions like a single neuron of an artificial neural network [5], in which a neuron gets a signal pulse from the previous neuron associated with a given weight. The perceptron receives many such inputs $x_1, x_2, \ldots, x_N$ which are multiplied with the corresponding weights $w_1, w_2, \ldots, w_N$ and summed. The weights signify the role of the corresponding feature of the dataset.

The crucial mathematical expression of perceptron learning algorithm is shown in (2).

$$y = sign(\sum_{i=1}^{N} w_i x_i + w_0) \quad (2)$$

The y indicates the class assigned to the point x_i by the preceptron and w_0 is the bias. In each iteration the weights are updated based on rule in (3).

$$w_j^{(k+1)} = w_j^{(k)} + \eta(y_i - \hat{y}^{(k)})x_{ij} \quad (3)$$

Where, $w_i^{(k)}$ is the weight corresponding to the i^{th} input at the k^{th} iteration. η is called as the learning rate and w_{ij} is the weight associated with the j^{th} attribute of training parameter $\boldsymbol{x_i}$.

The perceptron was trained for 123 iterations with a learning rate $\eta = 1.0$. The classification accuracy was 98.91% as shown in Table 3. For plotting decision boundaries in 2D, the classifiers have been trained on the data with attributes variance and skewness alone, with same hyper-parameters as before. Variance and skewness are chosen because, we found that these attributes hold most of the information in the data. Hence all the decision boundary plots correspond to classifiers trained on 2D data. The decision boundary learnt by the classifier trained on 2D data is shown in Fig. 6.

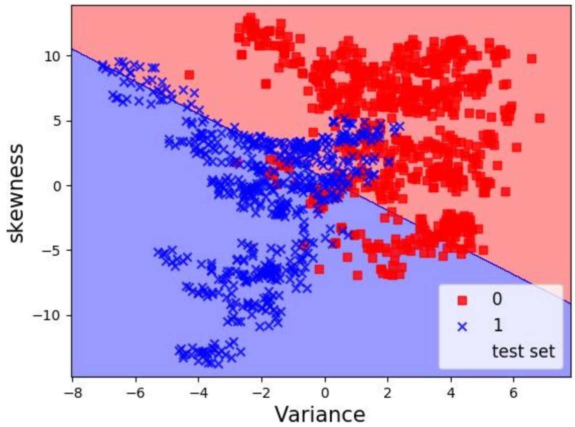

Fig. 6. The decision boundary learnt by Perceptron model.

C. Logistic Regression Classifier

Unlike the perceptron algorithm, the logistic regression can handle data that is non-linearly separable. It however works better for linearly separable data.

The logistic function (4) gives the probability that a given data point x belongs to a class k. This is done while minimizing the cost function using gradient descent (5).

$$P(Y_i = k \mid X_i) = \phi(z) = \frac{1}{1+e^{-z}} \quad (4)$$

$$J(w) = \sum_{i=1}^{n} -\log(\phi(z^{(i)}) - (1-y^{(i)})\log(1-\phi(z^{(i)})) \quad (5)$$

where $y^{(i)}$ denotes the actual class corresponding to a given input x_i and $\phi\ (z^{(i)})$ denotes output of the classifier.

The logistic regression was run on the training data with a inverse regularization length(C) of 1.0. The classification accuracy was 99.27% as shown in Table 3. The decision boundary learnt by the classifier (trained on 2D data) is plotted using the features variance and skewness, as shown in Fig. 7.

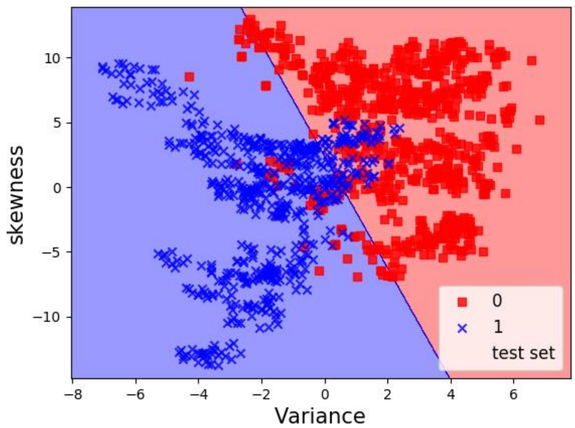

Fig. 7. The decision boundary learnt by Logistic Regression Model.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

D. Decision Tree Classifier

Like the logistic regression, Decision tree classifier can handle data that is non-linearly separable. The intermediate variables of this algorithm will help us calculate the relative importance of each of the parameters and can help in dimensionality reduction.

The algorithm makes a list of attributes in the input dataset, and ranks them based on reduction in entropy (6).

The entropy of an attribute x is given by:

$$H(x) = -\sum p(x)\log p(x) \quad (6)$$

The decision tree algorithm was run on the training data with pre-pruning depth of 6. The classification accuracy was 98.91% as shown in Table 3. The decision boundary learnt by the classifier trained on 2D data as shown in Fig. 8.

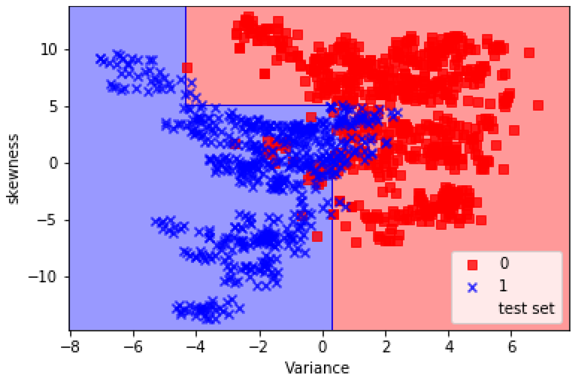

Fig. 8. The decision boundary learnt by Decision Tree Classifier.

E. Artificial Neural Network Classifier

An appropriately designed Multi-Layer Perceptron(MLP) network is said to model any arbitrary function. As the banknote dataset is not linearly separable, we use MLP for the task. The MLP used has 2 hidden layers of 5 neurons each. Each neuron in the MLP output is given by (7).

$$y = \sum_{i=1}^{K} w_i x_i + b \quad (7)$$

where y is the output K denotes number of perceptrons in the previous layer and w_i denote the weights of links from i^{th} perceptron in the previous layer to the current perceptron. The input to the current perceptron is $y+b$, where b is a bias. The error in output is given by (8).

$$E = \frac{1}{2}\sum_{i=1}^{N}\left\|o_i - t_i\right\|^2 \quad (8)$$

Where, o_i is the output for the i^{th} training sample and the corresponding target is t_i. N is the number of training examples. The weights are adjusted by back propagating the errors [6]. The back propagation algorithm is used to find a local minimum of the error function.

The gradient of the error function is computed and used to correct the initial weights. It is computed recursively.

Using the above algorithm a 6 layer MLP, with 6 neurons per layer was trained using the training data. The classification accuracy for this model was 100.00% as shown in Table 3. The decision boundary learnt by the classifier trained on 2D data, as shown in Fig. 9. This model performed better than the previous approaches and we achieved perfect classification.

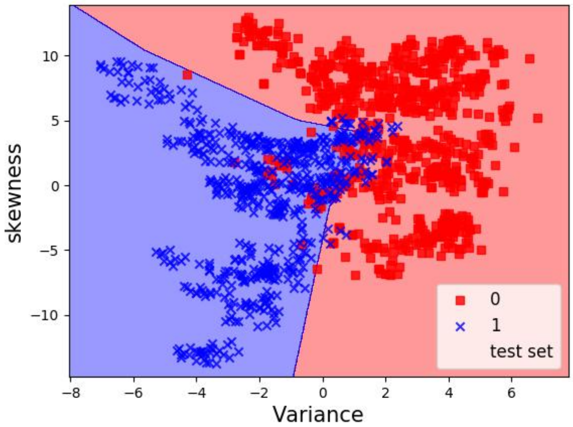

Fig. 9. The decision boundary learnt by ANN Classifier.

F. Support Vector Machine Classifier

SVM is a classifier that uses optimization techniques to maximize the performance, by maximizing the distance between support vectors. The kernel trick using Radial Basis Function(RBF)(12) has been applied to allow SVM to classify the non-linear banknote dataset [7].

The training of the SVM is purely a convex optimization problem. A quadratic objective function (9) needs to be minimized, subject to the linear constraint (10).

$$\left\|w\right\| = \frac{1}{2}ww^T \quad (9)$$

$$y_i(x_i w^T + b) \geq 1 \quad \forall i = 1 \; to \; N \quad (10)$$

Where, y_i is the class label of the point x_i and N is the total number of points in the input space.

Integrating the constraints to Lagrangian form, the equations obtained would be as in (11).

$$J(w,b,\alpha) = \frac{ww^T}{2} - \sum_{i=1}^{N}\alpha_i y_i(x_i w^T + b) + \sum_{i=1}^{N}\alpha_i, \alpha_i \geq 0 \; \forall i \quad (11)$$

RBF kernel function can be used for parameter selection [7].

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) = \exp(-\gamma\left\|x_i - x_j\right\|^2), \gamma > 0 \quad (12)$$

The SVM was run on the training data with the RBF kernel. The classification accuracy was again 100% as shown in Table 3. We achieved a perfect classification.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The decision boundary learnt by the classifier trained on 2D data, as shown in Fig. 10.

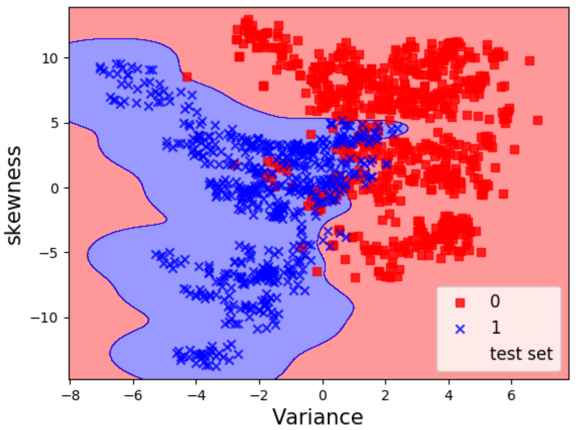

Fig. 10. decision boundary learnt by SVM.

G. K-Nearest Neighbour Classifier

KNN is a lazy learning algorithm that goes back to the training data to classify unseen data. Therefore it is not limited to linearly separable data, hence we have chosen to apply KNN on our data. The basic framework of KNN is as follows

1. Choose the number of k and a distance metric.
2. Find the k nearest neighbors of the sample that we want to classify.
3. Assign the class label by majority vote.

The KNN classifier was run on the training data with number of neighbours k=5. The classification accuracy was **100%** as shown in Table 3. The decision boundary learnt by the classifier trained on 2D data, as shown in Fig. 11. Here too we achieved perfect classification.

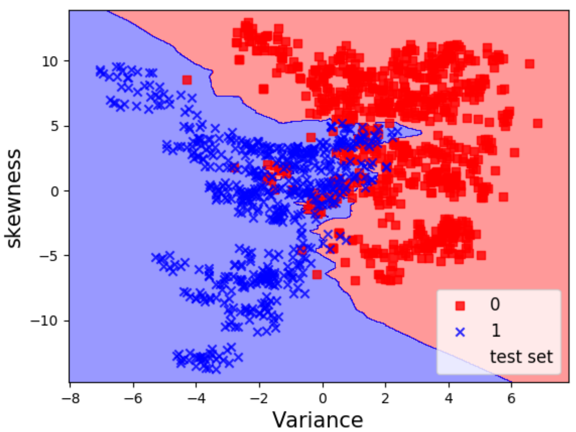

Fig. 11. The decision boundary learnt by KNN model.

The decision tree algorithm, and the PCA on the dataset, determined that the parameters Variance and Skewness carry the maximum information. If we could use lesser number of parameters while achieving similar performance it would be beneficial in a real time implementation scenario. Hence all the trials with all the six algorithms were repeated using just these two parameters.
The results are given in Table 3, we see that the performance using just these two parameters is poorer than the case where all four parameters are used for classification. Further discussion is based on using all four parameters for the feature vectors.

Table. 3. Showing Performance Of Various Classifiers On Banknote Dataset.

Classifier	VAR,SKE,KUR,ENT	ACCURACY
Perceptron	98.91%	82.18%
Logistic Regression	99.27%	86.18%
ANN	**100.00%**	92.36%
Decision Tree	98.91%	91.64%
SVM	**100.00%**	92.00%
KNN	**100.00%**	92.36%

V. RESULTS AND DISCUSSION

Classification models were built using Perceptron, Decision Tree, Logistic Regression, Artificial Neural Network, Support Vector Machine and K Nearest Neighbour algorithms.

The perceptron algorithm was trained for 123 iterations with a learning rate $\eta = 1.0$. The classification accuracy was 98.91\% as shown in Table 3. The decision boundary learnt by the classifier trained on 2D data is shown in Fig. 6. The logistic regression was run on the training data with a inverse regularization length(C) of 1.0. The classification accuracy was 99.27% as shown in Table 3. The decision boundary learnt by this classifier is as shown in Fig. 7. The decision tree algorithm was run on the training data with pre-pruning depth of 6. The classification accuracy was 98.91% as shown in Table 3. The decision boundary learnt by the classifier is as shown in Fig. 8. A MLP with 6 layers where each layer has 6 neurons was trained using the training data. The classification accuracy for this model was 100.00% as shown in Table 3. The decision boundary learnt by the classifier is as shown in Fig. 9. This model performed better than the previous approaches and we achieved perfect classification. The SVM algorithm was run on the training data with the RBF kernel. The classification accuracy was again 100% as shown in Table 3. We achieved a perfect classification. The decision boundary learnt by the classifier is as shown in Fig. 1. The KNN classifier was run on the training data with number of neighbours k = 5. The classification accuracy was 100% as shown in Table 3. The decision boundary learnt by the classifier trained on 2D data, as shown in Fig. 11. Here too we achieved perfect classification.

VI. CONCLUSION

Automatic detection of counterfeit banknotes with perfection is an absolute necessity. The perceptron, decision tree, logistic regression, ANN, SVM and KNN algorithms were employed and fine tuned. Perfect classification (100%) was achieved using ANN, SVM and KNN. The logistic regression followed by decision tree and simple perceptron have also performed well by detecting fake notes at 99.27%, 98.91% and 98.91% respectively. Optimizing or speeding up the classifier by using a subset of features, namely variance and skewness, however did not aid in improving the classification accuracy.

In situations where the volume of transactions (counting and verification) is huge, as in a central bank, where speed is of essence, it may be appropriate to use just the two features (VAR and SKE) for classification though at reduced accuracy. An alternative and more appropriate solution though would be to implement any of the algorithms ANN, SVM or KNN on hardware in order to achieve speedup while maintaining accuracy.

ACKNOWLEDGMENT

Malladi Tejasvi Author thanks his parents **Dr. Malladi Surya Prakasa Sastry** and **Malladi Subba Lakshmi** for their constant encouragement and support to carry out this research work. Malladi Tejasvi Author also thanks **VIT University and its management** for constant support and guidance for this research work.

REFERENCES

1. Lichman, M., 2013. UCI machine learning repository.
2. Kitagawa, R., Mochizuki, Y., Iizuka, S., Simo-Serra, E., Matsuki, H., Natori, N., and Ishikawa, H., 2017. "Banknote portrait detection using convolutional neural network". In Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on, IEEE, pp. 440–443.
3. Kang, K., and Lee, C., 2016. "Fake banknote detection using multispectral images". In Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference on, IEEE, pp. 1–3.
4. Uddin, M. S., Das, P. P., and Roney, M. S. A., 2016. "Image-based approach for the detection of counterfeit banknotes of bangladesh". In Informatics, Electronics and Vision (ICIEV), 2016 5th International Conference on, IEEE, pp. 1067–1072.
5. Hopfield, J. J., 1988. "Artificial neural networks". IEEE Circuits and Devices Magazine, 4(5), pp. 3–10.
6. Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al., 1988. "Learning representations by back-propagating errors". Cognitive modeling, 5(3), p. 1.
7. Lin, S.-l., and Liu, Z., 2007. "Parameter selection in svm with rbf kernel function". Journal-Zhejiang University of Technology, 35(2), p. 163.

AUTHORS PROFILE

Malladi Tejasvi, is pursuing his final year of B.Tech in Computer Science and Engineering at VIT University, Chennai. He is the first ranker in his batch. His research interests include Machine Learning, Cyber Security and Image Processing.