[1] **Himanshu**
[2] **Niraj Singhal**
[3] **Anuradha Singh**
[4] **Neeraj Pratap Singh**
[5] **Pradeep Kumar**

Identification of Counterfeit Currency using Machine Learning and Knowledge Discovery

Abstract: - Today, every major economy must deal with the problem of counterfeit money. Counterfeit currency is a currency that is produced without the state's or governments legal approval. A portion of the negative cultural repercussions remembers a drop in the worth of genuine cash and an expansion in costs as more cash circles in the economy. That is a reason why governments have used fictitious currencies to wage economic warfare against one another. As a result, we must implement counter-measures that will aid in the prevention of this threat. It is feasible to create high-quality counterfeit banknotes that are difficult to recognize from real notes using computers and technology. In reality, several counterfeit notes were confiscated, many of which replicated many of the security measures found in actual currency notes. As a result, we must develop new approaches to assist consumers in more accurately and comfortably identifying counterfeit cash notes. Knowledge database discovery and machine learning approaches can be used to create tools that can assist with this endeavor. We can train computers to recognize patterns or traits that help them distinguish between real and counterfeit cash. Therefore, the main goal of this research is to create a model that can be utilized to identify fake currency with the least amount of classification mistakes after being trained using pertinent.

Keywords: Support Vector Classifier, Gradient Boosting Classifier, K-Nearest Neighbors Classifier, Data Exploration, Exploratory Visualization

I. INTRODUCTION

The detection of counterfeit money notes is critical to the economy's integrity. There has been an increase in the usage of machine learning models for identifying counterfeit currencies utilizing data mining and knowledge discovery in recent years. Because identifying phony money is extremely difficult for humans, automated technologies for detecting false cash are essential. Fake cash is money created without the authority of the government; creating it is a serious offense [1]. The advancement of color printing technology has significantly boosted the pace of counterfeit currency note production on a wide scale. Previously, printing could only be done in a print shop, but now anyone with a low-cost laser printer can print a currency note with precision. As a result, the usage of counterfeit notes in place of legal ones has increased dramatically. It is the most serious issue confronting many countries, including India. Though banks and other major organizations have installed automatic technology to identify counterfeit money notes, the common person finds it difficult to discern between the two [2].

The most danger in the banking sector is the creation of counterfeit cash. UV light is commonly used to prove authenticity. Note value, ink smudge, Security thread, serial number, Intaglio printing, watermark, reserve bank number panel, LD mark, Topography, Micro-lettering, and numbers & alignment are all examples of features on banknotes are the key elements used to detect counterfeit cash [3]. A watermark, ink smear, security thread, topography, numbers & position, and tiny writing are all crucial elements. However, for machine-based assessment, researchers often do the following procedures. The procedures are as follows [1].

a. Pre-processing
b. Segmentation of Image.
c. Extraction of features.
d. Feature classifiers or matching

Machine Learning approaches to aid in the development of apps that facilitate currency identification using automated systems and algorithms. Machine Learning will analyze real-world characteristics by utilizing pattern

[1] Assistant Professor, Meerut Institute of Technology, Meerut, India.

[2] Director, Sir Chhotu Ram Institute of Engineering & Technology, C.C.S. University, Meerut, India

[3] JSS Academy of Technical Education, Noida, India.

[4] Professor, Meerut Institute of Technology, Meerut, India.

[5] JSS Academy of Technical Education, Noida, India.

recognition and knowledge-finding features. This study seeks to provide a supervised paradigm using related set theory, which will be useful in detecting faked datasets with relatively few categorizing flaws [4]. As a result, another term for the categorizing model organized as data, comprising of qualities and labels for the bills indicating whether they are fraudulent or real, is used. Furthermore, it determines decision boundaries that divide samples into two groups.

Printed money comes in a variety of denominations and can appear as coins or banknotes, each with a specific value. Coins are frequently less valued than banknotes. The need for banknote identification devices has increased as a result. There are several banknote recognition equipment available, however, while scanning a note for legitimacy, they have a diagramming limit. Due to recent technical advancements, printing fake currency is now easier than ever thanks to copy machines and scanners [5]. It could be challenging to tell which notes are real and which are false when there are several of them that appear alike. A system that enables a user to determine if a specific collection of notes is authentic based on its properties is therefore required. We were able to develop and propose a machine learning-based banknote authentication solution as a result of this necessity [6].

Today, counterfeit money is a major problem for governments all over the world. To determine if a coin is fake or not, the author is experimenting with several machine-learning algorithms. Practically every area, including healthcare prediction, cyberattack prediction, credit card fraud detection, and a plethora of others, has already seen the value of machine learning algorithms in action. The author advises using machine learning to identify fake currency as a result [7].

In India, daily financial transactions have grown in recent years as a result of an increase in daily transactions. The benefit is that fake money of Rs. 50, 100, 500, and 1000 is produced, but since demonetization, the number of counterfeit notes of the new Rs. 50, 200, 500, 2000 has substantially increased, hindering the nation's economic development [8]. In India, only the Reserve Bank of India (RBI) has the right to issue banknotes. The RBI deals with the problem of fake money every year. It is important to address the growing problem of counterfeit money in India. Indian rupee counterfeit note production has considerably expanded in recent years. The government has not given its approval for counterfeit money [9].

II. LITERATURE SURVEY

Machine learning algorithms using massive amounts of data as inputs will be used to identify counterfeit banknotes. The KNN method is an effective data analysis tool. These kinds of analyses can be quite beneficial for obtaining implicit information.

In this work, we evaluate the most recent research on the classification and clustering of sequential data. This research study also goes over some of the most common applications for cash detection. We employ three distinct types of algorithms, the KNN technique, the gradient method, and the support vector approach, coupled with the notion of machine learning, to identify currency on a big scale and to liberate India from such malpractice.

In a study, *Ying Li Tian* identifies bogus notes for the blind using image processing and segmentation. MATLAB software is used to extract various properties of money notes. This improves simplicity as well as high-performance speed. In a publication [10], Deep learning using SVM and FNN (Feed Forward Neural Network) may detect fake notes, according to *Li Liu et al.* FNN is also employed in verification. It makes use of the max pool operation; if an image is extracted, it is then subjected to an augmentation process and then annotation; these allow database building; we then input the image in real-time via transfer learning by Alex-network; and finally, it is subjected to feature extraction., which compares real-time and database features. Finally, it guesses if the currency is false or real. *Renuka Nagpure and Shreya Shetty* [11], money duplication is a significant danger to the economy, and it is becoming a regular problem owing to sophisticated technology and laser printers; certain measures are being used to combat this. The register, watermarking, optically changeable ink, security thread, fluorescence, latent image, and identifying mark can all be used to detect cash. This is advantageous to the banking industry. In an article [12,] authors *Bo Tang and StevanKay* describe a unique form feature utilizing the angle distance approach. For input size detection and text classification, an automated feature selection and automated feature reduction technique are used. We are using three supervised learning approaches for this problem.

A. Support Vector Classifier

A hyperplane or group of hyperplanes in a high- or infinite-dimensional space can be created using the supervised learning technique known as SVM for classification, regression, and other tasks. Because the wider the margin, the less the classifier's generalization error, the hyperplane with the greatest distance to the closest training data point

of any class (referred to as functional margin) provides an acceptable separation. Because the present problem is a binary classification problem with a modest number of features in comparison to the amount of input, the Support Vector Classifier will work.

- It works well in three-dimensional areas.
- When a suitable kernel is chosen, it performs well with non-linear decision boundaries.
- It is especially helpful when there are more dimensions than samples.
- By incorporating a portion of training points (referred to as support vectors) in the decision function, it preserves memory.

B. Gradient Boosting Classifier

A Gradient boosting model is made up of weak learners that work together to build an extremely robust model. A weak learner is a rudimentary prediction model that outperforms random chance by a small margin. Combining rough and partly erroneous rules, it creates a highly accurate prediction rule. Decision trees, which are a flowchart of Yes/No questions, are used as a weak learner or prediction rule. The weak learners are introduced progressively, one at a time, and by weighing the observations, more weight is given to difficult-to-classify circumstances, and less weight is given to those that have previously been handled effectively. Gradient Boosting Classifier will work for the current situation since the data is arranged in such a manner that it is suitable for asking layered questions in decision trees.

- This classifier is simple to understand and explain.
- It is capable of capturing complicated non-linear function relationships.
- Extremely adaptable and easily customizable to meet a variety of practical requirements.
- Combines the results of numerous weak learners to create a more robust model.

C. K-Nearest Neighbors Classifier

KNN classifies a given data point by examining its neighbors and giving them weights so that the nearest neighbors have a stronger influence on class selection. It is a type of lazy learner. The distance can be determined using Murkowski, Euclidean, and other methods. KNN is referred to as a lazy learner since it categorizes test point sets based only on their closest neighbors and does not create a model on training data. Because our dataset is limited, the K-Nearest Neighbor Classifier will suffice in this scenario. As a result, making predictions using KNN would be quick.

- This classifier is easy to set up.
- Adaptable in terms of feature or distance selection.
- Handles multi-class scenarios with ease

III. METHODOLOGY

3.1. Data Exploration

The banknote authentication dataset contains five features, as previously stated: variance, skewness, kurtosis, entropy, and class. The first four continuous features retrieved by wavelet processing from a dollar bill image will be used to train our model. In this example, the fifth attribute is the class label, which might be 1 (authentic) or 0 (fraud). This dataset contains 1372 samples, 762 of which are false notes and 610 of which are authentic notes. The UCI Machine Learning Repository dataset is lacking names for its five properties, which must be filled up manually for a more accurate interpretation. Table 1 is a tabular representation of the first few records.

	Variance	**Skewness**	**Kurtosis**	**Entropy**	**Class**
0	3.72160	8.7661	-2.7073	-0.45699	0
1	4.64590	8.2674	-2.5586	-1.56210	0
2	3.96600	-2.5383	1.8242	0.11645	0
3	3.55660	9.6228	-4.1112	-3.69440	0
4	0.42924	-4.3552	4.6718	-0.99880	0

3.2 Exploratory Visualization

A dataset could have at least one character with values that are near to a single number, but it will also likely include a sizable number of values that are higher or lower. Such value distributions make algorithms susceptible, and if

the range is not adequately normalized, the algorithm will fail or perform poorly. To determine the presence of any such outliers, we plot a histogram for each characteristic. The histograms are depicted in Figure 1.

These distributions highlight the following points:

(i) In this dataset, there are no characteristics for which a sample has a very big or very small value in contrast to the other sample values.

(ii) We don't need to do a log transformation because there are no such outliers.

(iii) The 'variance' and skewness qualities have a very even distribution.

(iv) Kurtosis distribution is positively skewed, with most records having a kurtosis value of less than 3.

(v) Because of the negatively skewed distribution of entropy, more samples have greater entropy, resulting in more images with higher contrast.

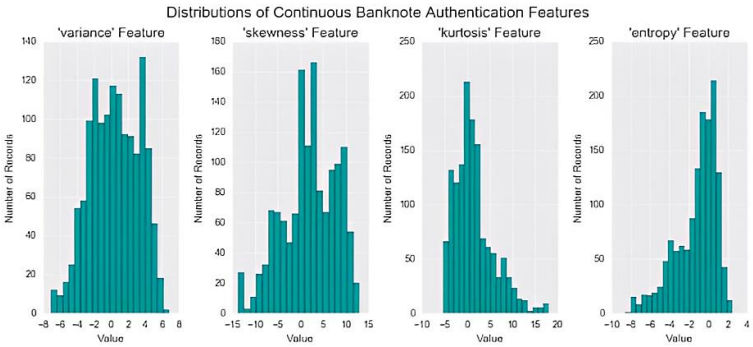

Figure 1: Distributions of features in the banknote authentication dataset

The next analysis would be to produce six classification scatter plots, each with two of the four numerical characteristics and distinct colors for items belonging to class 1 and class 0. This would aid us in identifying the pair of attributes that most clearly distinguishes the two classes. Class 1 (genuine note) is colored green in the scatter plots shown in Figure 2, whereas class 0 (false note) is colored red.

We note the following points from the scatter plots:

(i) Skewness, the Variance pair, most firmly distinguishes the two classes of notes, and a piecewise linear boundary will yield satisfactory results.

(ii) Entropy, the Kurtosis pair is the trait that most firmly distinguishes the two classes of notes, and algorithms will struggle to conduct excellent classification if we depend just on these two features.

(iii) Variation is a key trait that separates real from counterfeit notes. Real money bills have a low variance, but counterfeit bills have significant volatility.

(iv) Entropy is the least essential attribute in terms of class separability.

For many pairings of traits, the scatter plots demonstrate significant separability between the two classes. So it looks like our features are fairly successful at distinguishing between two sorts of banknotes, and with the right algorithm, we should be able to reach high forecast accuracy.

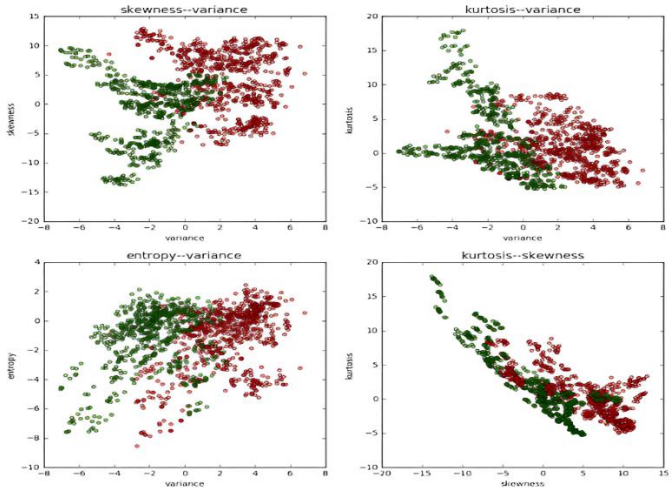

Figure 2: Scatter plots of all pairs of continuous features

3.3. Algorithms and Techniques

Three supervised learning methods are utilized for this aim. These algorithms were picked in such a manner that their methodologies differed significantly from one another, allowing us to cover a wide spectrum of possible methods. We will compare the outcomes of the three strategies listed below to our benchmark model.

(i) Support Vector Classifier
(ii) Gradient Boosting Classifier
(iii) K-Nearest Neighbors Classifier

3.4 Data Preprocessing

Data preprocessing is required before feeding data into an algorithm to get it into excellent form, eliminate any anomalies, or adjust any properties. We highlighted the following aspects of our data in the data visualization section.

(i) The dataset has no missing values.
(ii) Because all of the input characteristics are continuous and have distinct ranges, we must apply normalization to scale all features in the 0-1 range.

3.4.1. Normalization

Based on the data analysis, we determine that we need to scale the four continuous characteristics to bring them down into the 0 to 1 range. The classifier will then handle all characteristics equally. To normalize our characteristics, we apply the following formula:

*X*norm=Xcurrent-Xmin / Xmax-Xmin

We perform scaling so that the supervised learning algorithm is not biased toward any feature. The first five samples after normalization are reproduced here.

	Variance	**Skewness**	**Kurtosis**	**Entropy**
0	0.779004	0.849643	0.116783	0.746628
1	0.845659	0.830982	0.131804	0.654326
2	0.796629	0.426648	0.320608	0.796951
3	0.767105	0.881699	0.055921	0.460440
4	0.541578	0.358662	0.434662	0.697362

3.5 Implementation

Now that our data has been normalized, we should divide it into training and test sets. After that, we utilize the training set to train our classifier, which is then tested on the test set. The outcomes will be compared to the reference model.

3.5.1 Splitting data

We divided the samples to put the trained model to the test with samples it had never seen before. This ensures that the model derives categorization patterns from the training data rather than memorizes them. We use the train_test_split function from the sklearn library's cross-validation module to divide our data for the specified issue. This function currently serves two key purposes.

(i) It shuffles the dataset such that each class has nearly the same number of instances in both the training and test sets.
(ii) It does the split after shuffling. We can select how much of the total data to utilize for training or testing.

Following that, four lists are returned: training set input features, test set input features, training set target labels, and test set target labels. We split the data so that 60% of the samples are in the training set and 40% are in the test set.

3.5.2 Creating Training and Prediction Pipeline

We develop the 'train_split' method, which accepts the following arguments as inputs: learner, sample_size, X_train, Y_train, X_test, y_test. It returns the test and training sets' accuracy and F-beta values. By adjusting the learner to training data of the size supplied by "sample_size," the function calculates training time. The predictions from the test set and 300 samples from the training set are then used to calculate the time spent on prediction.

The accuracy and f-beta scores are then calculated for the training (300 samples) and test sets. To improve prediction visualization, we additionally develop a confusion matrix. We train the three classifiers with sample

sizes of 5%, 20%, and 100% of the training data to see how performance varies with the train set size. Figure 3 depicts bar graphs of the three classifiers' accuracy and f-scores on training and test sets.

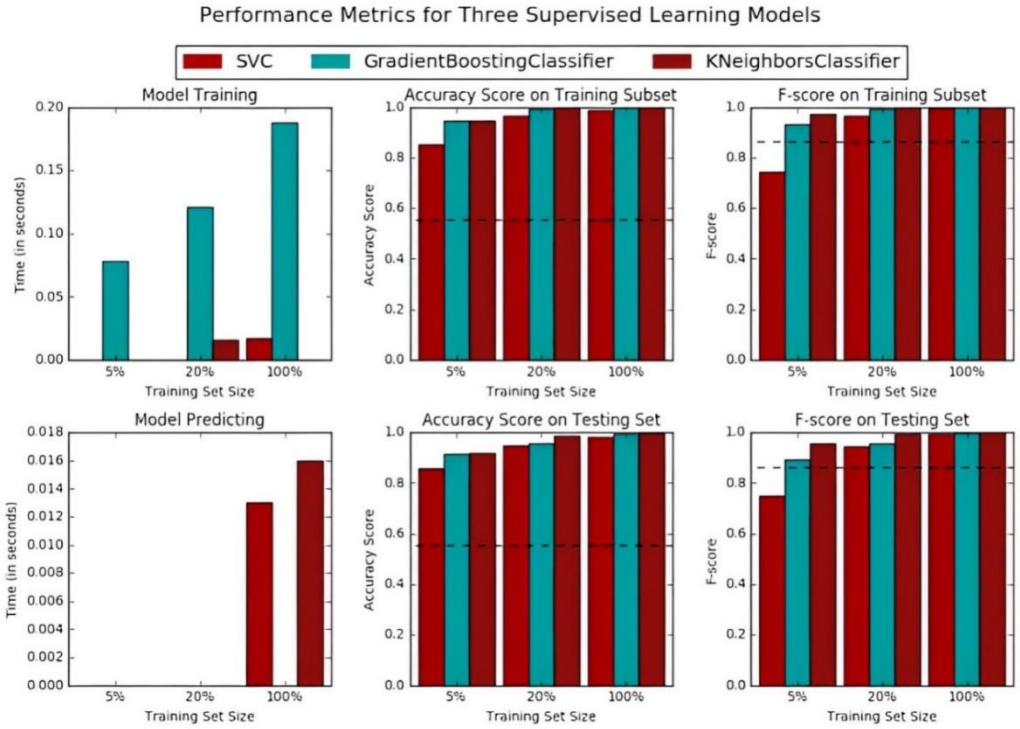

Figure 3: Performance Metrics for three supervised learning models

When we employ the whole train set for training, we receive the following results.

	Support Vector Classifier	**Gradient Boosting Classifier**	**K-Nearest Neighbors Classifier**
A training set of accuracy	0.9967	1.0	1.0
Testing set of accuracy	0.99	0.9955	0.9992
A training set of f-score	0.9951	1.1	1.0
Testing set of f-score	0.9924	0.9963	0.9993

The Confusion matrix of test data prediction is also presented below for all three classifiers.

Support Vector Classifier	
284 (TN)	11 (FP)
0 (FN)	254 (TP)

Gradient Boosting Classifier	
293 (TN)	2 (FP)
1 (FN)	253 (TP)

K-Nearest Neighbors Classifier	
294 (TN)	1 (FP)
0 (FN)	254 (TP)

The above visualization and stats highlight the following points:

(i) All three algorithms outperform the test set necessary for a counterfeit money note-detecting application.

(ii) With a recall of 1.0, the Support Vector Classifier and K-Nearest Neighbors Classifier algorithms detect all counterfeit money notes.

(iii) The K-Nearest Neighbors Classifier and Gradient Boosting Classifier have the best precision and produce the fewest false positives.

(iv) On the test set, the K-Nearest Neighbors Classifier achieves an accuracy of 0.9992 and an F-score of 0.9993. It accurately identifies all test samples, except for one false positive.
(v) On both training and test sets, the K-Nearest Neighbors Classifier outperforms the other two techniques for all training sample sizes.
(vi) Based on the visualizations and evaluation metrics, it is obvious that the K-Nearest Neighbors Classifier is the clear winner; however, other methods require optimization to perform as well.
(vii) While gradient boosting necessitates some training time and the K-Nearest Neighbors Classifier necessitates testing time, the overall time required is negligible, on the order of a fraction of a second.
(viii) The scatter plots we created previously predicted such outstanding outcomes.

We selected the K-Nearest Neighbor Classifier to address this problem since it generated the best results and successfully classified all bogus money notes. Even if the K-NN Classifier produces such good results that further refining is unnecessary, we will still perform a Grid Search and make an effort to optimize the settings to make sure that changing the parameters won't result in an improvement.
We want to optimize the following settings for the KNN algorithm:
(i) ‘n_ neighbors' specifies the number of neighbors to utilize. Six values are considered [1, 2, 3, 4, 5, 6].
(ii) ‘Weights': the weight function that is utilized in prediction. We choose two options ['uniform' and 'distance']. The 'uniform' value equally weights each point in the neighborhood, but the 'distance' value gives closer neighbors greater impact.
(iii) ‘Algorithm': The algorithm is used to calculate the nearest neighbors. We consider suitable values to be ['ball_tree', 'kd_tree'].

IV. Result And Implementations

In terms of accuracy and f-beta scores, the KNN classifier beat the other two classifiers. As a result, it is our preferred model as a solution to the situation at hand. We picked KNN as the final model based on the following factors.
Starting, the KNN classifier outperforms the other two classifiers in terms of accuracy and f-beta scores on both the training and test sets. On the training set, the accuracy and f-beta scores are both 1.0, whereas, on the test set, they are 0.9992 and 0.9993, respectively. It correctly classified every phony note in the test set, misclassifying just one genuine note. Second, on varying training set sizes, the model consistently outperformed the other two classifiers. The prediction accuracy on both training and test data approached 90% despite the training set is just 5% of its real size. This represents how difficult the algorithm is. Third, the KNN results were so good that even fine-tuning the model's parameters with grid search produced identical results.
The grid search results show that the optimized model's 'n_neighbors' parameter is set to 1. The number of closest neighbors who cast ballots for the predicted outcome is shown by the variable "n_neighbors." However, we originally believed that the value of 'n_neighbors' was 5. This prompted us to investigate how the number of neighbors in this situation affects KNN performance. Figure 5 shows the outcomes. We discovered that the F-beta score and accuracy for the first 10 nearest neighbors are identical. This again displays the KNN Classifier's adaptability to the problem at hand. What therefore should the value of "n_ neighbors" be in the model for the complete solution? Although Option One seems simple, our model becomes more sensitive to new input as a result. Some sample labels could change as a result of a single noisy point or outlier. The model is complicated by several 5s or higher. As a consequence, we should select a few 3 or 4s. However, because 4 is an even number, we will set 'n_ neighbors' to 3 in our final KNN classifier model. As a result, we may infer that the model is somewhat resilient because it delivers unexpectedly identical results on varied training set sizes and different values of its parameter 'n_ neighbors'. The model's persistence increases our confidence in its selection as the ultimate answer.
On many levels, the model's output may be trusted. For starters, the test set was distinct from the training set. As a result, during the testing phase, the model was tested on unseen samples. Second, we did a grid search using cross-validated randomized folds to exclude the possibility of overfitting and obtained comparable findings. Third, we obtained comparable findings with various models. So, the results are reliable.
Finally, we feel the model is logical and meets our expectations for a decent solution because it works very well and is quite resilient. The model is also highly rapid, giving accurate classification results in a fraction of a second.

V. CONCLUSION

To demonstrate how each algorithm performed on the test data, we created a Confusion Matrix for it. For the program to be successfully deployed, any model must accurately classify all counterfeit money. KNN-Classifier properly detected all bogus notes and misclassified just one genuine note. A few false positives are acceptable in the context of this situation, but any false negative is not. So, using the model of our choosing, we were able to achieve our aim. Furthermore, we discovered that the KNN Classifier performs well in terms of training set size and parameters. This is required for every application with monetary consequences. In this study, we find an intriguing dataset that has the potential to address a real-world issue. The UCI Machine Learning Repository's Banknote Authentication dataset appears to be a great choice. The primary step in tackling a situation like this is to examine the data. We ran the statistical analysis and discussed the dataset's properties. We then created several visualizations that reflected the link between various characteristics and their capacity to classify. We scaled the dataset's continuous features from 0 to 1.

REFERENCES

[1] Arun Anoop M and Dr. K. E. Kannammal, “Fake currency detection: A survey”, Gedrag & Organisatie Review, Vol 9, Issue 04, pp 622-638, December 2020.

[2] Akash Rana and Avanish Kumar, “Detection of Fake Currency using Machine Learning Technique”, International Journal of Creative Research Thoughts (IJCRT), Vol 9, Issue 5, pp 8-15, 2021.

[3] Surya, S and G. Thailambal. "Comparative Study on Currency Recognition System Using Image Processing." International Journal Of Engineering And Computer Science, Vol 3, Issue 08, 2014.

[4] P B V Rajarao and B S N Murthy, “Evaluation of Machine Learning Algorithms for the Detection of Fake Bank Currency”, Journal Of Algebraic Statistics, Vol 13, Issue 2, pp 3680-3688, 2022.

[5] D.Alekhya and G.DeviSuryaPrabha , G.VenkataDurgaRao, ”Fake Currency Detection Using Image Processing and Other Standard Methods”, International Journal of Research in Computer and Communication Technology, Vol. 3, Issue 1, January 2014.

[6] Arun Anoop M, Poonkuntran S,”Lora Approach For Image Forgery Detection And Localization In Digital Images”, International Journal of Social & Scientific Research, Vol 04, Issue 3, 2019.

[7] Pushpa R N and Aditya Aithal H K,” Detection of Fake Currency using Machine Learning”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 11, Issue 4, pp 4415-4421, 2023.

[8] Verma, Vivek K., Anju Yadav, and Tarun Jain. "Key Feature Extraction and Machine Learning-Based Automatic Text Summarization." Emerging Technologies in Data Mining and Information Security. Springer, pp 871-877,2019.

[9] P. Julia Grace and A. Sheema, ”A Survey on Fake Indian Paper Currency Identification System”, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, Issue 7, July 2016.

[10] Lohweg, Volker, et al. "Banknote authentication with mobile devices." IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2013.

[11] Renuka Nagpure, Shreya Shetty, Trupti Ghotkar, ChirayuYadav, SurajKanojiya, ”Currency Recognition and Fake Note Detection”, International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, Issue 3, March 2016.

[12] RinkiRathee, ”Design of HSV Mechanism for Detection of Fake Currency”, International Journal of Emerging Technology and Advanced Engineering, vol. 6, Issue 7, 2016.

[13] EshitaPilania and BhavikaArora, ”Recognition of Fake Currency Based on Security Thread Feature of Currency”, International Journal Of Engineering And Computer Science, vol. 5, Issue 7, pp. 17136-17140, 2016.

[14] B. R. Kavya and B. Devendran, “Indian currency detection and denomination using SIFT,” International Journal of Science, Engineering and Technology Research, vol. 4, Issue 6, pp. 1909- 1911, 2015.

[15] A. Babar, S. Jawalekar, K. Yadav, and D. B. Salunke, “ Counterfeit currency detector,” International Journal of Technical Research and Applications, vol.3, Issue 3, pp. 106-108, 2015.

[16] Muhammad Sarfraz,"An intelligent paper currency recognition system", International Conference on Communication, Management and Information Technology (ICCMIT 2015),Elsevier, Procedia Computer Science 65, pp 538-545 ,2015.