JOURNAL OF DATA SCIENCE | Vol.2024:33
eISSN:2805-5160

Evaluating Machine Learning Algorithms for Fake Currency Detection

Keerthana SN[1], Chitra K[2]

[1,2] Dayananda Sagar Academy of Technology and Management Bangalore, Karnataka 560082 India

Email: keerthanasn1983@gmai1.com[1] , Chitra-mca@dsatm.edu.in[2]

Abstract

Currency is a critical asset in any economy, yet it is vulnerable to counterfeiting, undermining its value and disrupting economic stability. Counterfeit currency is particularly prevalent during economic transition, such as demonetization, as fake notes are circulated to mimic real currency. Due to the subtle similarities between genuine and fake notes, distinguishing between them can be challenging. Consequently, financial institutions like banks and ATMs require robust automated systems to accurately detect counterfeit currency. In this study, we evaluate the effectiveness of six supervised machine learning algorithms—K-Nearest Neighbor, Decision Trees, Support Vector Machine, Random Forests, Logistic Regression, and Naive Bayes—in detecting the authenticity of banknotes. Additionally, we examine the performance of LightGBM, a gradient-boosting algorithm, in comparison to these traditional methods. Our findings contribute to developing reliable, automated systems for counterfeit detection, and enhancing financial security.

Keywords

Counterfeit Detection, Machine Learning, Banknote Authentication, Currency Recognition, LightGBM

Introduction

Banknotes are one of the nation's most valuable assets, essential for daily transactions and economic stability. However, counterfeiting has become a significant issue as technology enables criminals to produce fake currency that closely resembles genuine notes, undermining the economy and contributing to illicit activities (Aoba, Kikuchi, & Takefuji, 2003). While counterfeiting was not a major problem in the early 20th century, technological advancements have made it increasingly difficult to distinguish between real and fake currency (Hassanpour & Hallajian, n.d.). In response, governments have introduced security features in banknotes to help identify genuine notes. However, counterfeiters continue to improve the accuracy of fake notes, making it challenging even for experts to differentiate between authentic and counterfeit currency (Prime & Solomon, 2010).

Submission: 27 June 2024; **Acceptance:** 28 October 2024

To address this challenge, financial institutions, such as banks and ATMs, require automated systems capable of reliably detecting fake currency. Machine learning offers a promising solution for this, as it can leverage image processing techniques and advanced algorithms to distinguish between real and fake banknotes with high accuracy. Supervised machine learning (SML) techniques, which have been successfully applied in various fields, including medical diagnosis and financial analysis, are particularly suitable for classification tasks in currency authentication (Desai et al., 2014). By training on annotated datasets of currency images, machine learning models can detect patterns unique to counterfeit notes and provide reliable classification results (Kumar & Dudyala, 2015).

Several approaches have been proposed for currency recognition. Aoba, Kikuchi, and Takefuji (2003) introduced a system using three-layered perceptron and radial basis function (RBF) networks for Euro banknote recognition. Similarly, Desai et al. (2014) employed multiple-kernel support vector machines (SVMs) to enhance counterfeit detection. In another work, Gigliarano, Figini, and Muliere (2014) addressed the challenge of choosing the best classifier when receiver operating characteristic (ROC) curves intersect, proposing a novel approach to model comparisons.

Feature extraction methods, such as texture-based analysis, have also been explored to improve classification accuracy (Hassanpour & Hallajian, n.d.). For example, Huang et al. (2004) applied support vector machines (SVMs) to enhance the explanatory power of the model, while Nastoulis et al. (2006) developed a probabilistic neural network that maintained recognition accuracy even with data errors.

In recent studies, additional machine learning algorithms have been tested for currency recognition. Omatu et al. (2007) examined the validity of neuro-classifiers and used local Principal Component Analysis (PCA) to reduce non-linear correlations among variables. Roy, Halder, and Garain (2010) focused on authenticating printing techniques using a pattern classification algorithm to verify the legitimacy of the text on currency notes. Additionally, recent advancements in machine learning, including LightGBM, Decision Trees, Random Forests, K-Nearest Neighbor, and Logistic Regression, have been applied to improve counterfeit detection accuracy by using well-structured datasets (Walia & Pal, 2015; Singh & Agarwal, 2018).

To create a robust system, this study proposes the use of multiple machine learning algorithms, including Decision Trees, Support Vector Machines, K-Nearest Neighbors, Random Forests, Naive Bayes, LightGBM, and Logistic Regression. The methodology encompasses data collection, feature extraction, model selection, and model evaluation. The performance of these models is highly dependent on the quality and diversity of the dataset, the feature selection process, and the precision of model training. Continued advancements in machine learning techniques and the adaptability of this system will play a crucial role in keeping up with evolving counterfeiting techniques. This system not only offers a reliable solution for financial institutions but also demonstrates the potential of machine learning in enhancing economic security by countering counterfeit currency.

Methodologies

Data Collection

Data collection involves gathering raw information from various sources to build a robust dataset for training machine learning models. In the context of banknote authentication, data may be sourced from images of real and counterfeit notes, capturing unique features that differentiate them (Aoba, Kikuchi, & Takefuji, 2003; Prime & Solomon, 2010). These datasets typically contain images, textures, and features essential for distinguishing genuine banknotes from counterfeit ones. High-quality and diverse data are crucial for improving model performance and reliability (Singh & Agarwal, 2018).

Data Preprocessing

Following data collection, data preprocessing is essential to prepare the dataset for analysis and modeling. This phase addresses issues such as missing data, noise, outliers, and inconsistencies within the data. Common preprocessing tasks include data cleansing, transformation, normalization, and feature selection. For example, eliminating irrelevant or redundant features and normalizing numerical data enhance model efficiency (Gigliarano, Figini, & Muliere, 2014). Handling missing values and converting data to a standard format also improves the model's accuracy in predicting counterfeit notes (Omatu et al., 2007).

Training and Testing

After preprocessing, the dataset is typically divided into two subsets: a training set and a testing set. The training set is used to teach statistical models by providing them with known inputs and corresponding outputs or labels (Desai et al., 2014). The model learns patterns and relationships within this data, which then applies when encountering unseen data. The testing set evaluates the model's performance by making predictions on new data and comparing these predictions to actual values, allowing for a realistic assessment of the model's accuracy and effectiveness (Nastoulis et al., 2006).

Data Modeling

Data modeling involves developing or selecting an appropriate model to capture and represent the patterns within the dataset effectively. In this study, models such as Support Vector Machines (SVM), Decision Trees, Random Forests, K-Nearest Neighbors, Naive Bayes, and LightGBM are considered for their efficiency in classification tasks (Huang et al., 2004; Kumar & Dudyala, 2015). Model complexity, performance optimization, and fine-tuning of parameters are all evaluated during this phase to ensure that the model achieves the highest possible accuracy in distinguishing between real and fake banknotes (Roy, Halder, & Garain, 2010).

Prediction

Once trained and validated, the model is ready for predictions on new, unseen data. This involves feeding the model with relevant features or variables extracted from new banknote images to generate a prediction output (Aoba, Kikuchi, & Takefuji, 2003). For classification tasks, such as determining the authenticity of currency, categorical labels (e.g., "Real" or "Fake") are used, with accuracy determined by the model's ability to generalize across diverse inputs. The model's chosen architecture, parameters, and training quality directly impact the reliability of these predictions in real-world applications (Hassanpour & Hallajian, n.d.).

This methodology outlines a systematic approach for building an accurate and robust machine learning system capable of detecting counterfeit currency, leveraging carefully structured data collection, preprocessing, modeling, and prediction phases. By using advanced algorithms and refining each step, the system ensures reliability in distinguishing authentic notes from counterfeit ones, supporting financial institutions in combating currency fraud.

Along with that, Figure 1 shows a typical workflow for building a machine learning model for predictive tasks, organized into several sequential stages that ensure data quality and model accuracy. Here's a detailed explanation of each stage:

1. Data Collection: The process begins with gathering raw data, which serves as the foundation for the machine learning model. This data may come from various sources and is collected in its original, unprocessed form.
2. Data Preprocessing: Once the raw data is collected, it undergoes preprocessing to clean and prepare it for analysis. This step involves tasks such as removing noise, handling missing values, normalizing data, and selecting relevant features. Preprocessing is crucial for enhancing data quality, which directly impacts the model's performance.
3. Preprocessed Data Storage: After preprocessing, the data is stored in a structured format, ready to be fed into the machine learning model. This dataset is now consistent, standardized, and contains only the most relevant information for analysis.
4. Model Selection: At this point, a suitable machine learning model is chosen based on the problem requirements and data characteristics. Model selection involves choosing the algorithm or technique (e.g., Decision Trees, SVM, Random Forests) that is most likely to yield accurate predictions.
5. Machine Learning (Training): The preprocessed data is then fed into the selected machine learning model. During this training phase, the model learns patterns and relationships within the data, allowing it to make predictions when presented with new data.
6. Prediction: Once trained, the model generates predictions based on input data. These predictions represent the model's attempt to solve the given problem, such as classifying an item or forecasting a trend.
7. Evaluation: The model's performance is evaluated to assess its accuracy and reliability. The evaluation phase involves comparing the model's predictions with actual outcomes to determine its effectiveness. Based on evaluation results, further adjustments to the model or preprocessing steps may be made to improve accuracy.
8. Feedback Loop: The diagram shows feedback loops connecting different stages, illustrating the iterative nature of machine learning. For instance, if the evaluation reveals performance issues, adjustments can be made to the preprocessing or model selection stages, leading to an improved model in the next iteration.

Overall, this workflow ensures that the machine learning model is built systematically, with each stage designed to refine data and model parameters to produce reliable and accurate predictions.

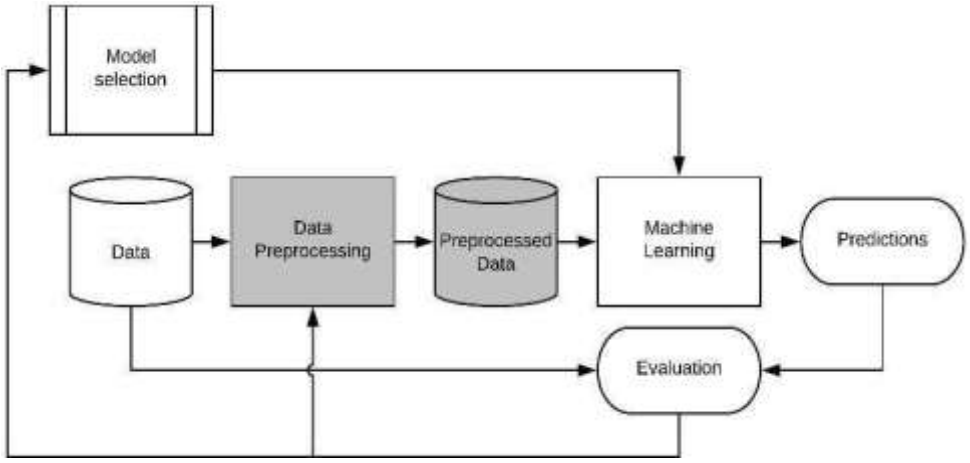

Figure 1. Machine Learning Model Development Workflow

Results and Discussion

In this study, we employed six machine learning algorithms—Support Vector Machine (SVM), Naive Bayes, k-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Logistic Regression—for classification and regression tasks. To evaluate model performance, we used metrics such as accuracy, precision, recall, and F-score, which helped us identify the features with the highest predictive potential.

Each classifier's performance was assessed based on these evaluation metrics. Additionally, we examined the impact of varying training data sizes on prediction scores to understand how the quantity of training data affects model accuracy. The classifier was trained on multiple subsets of the dataset, and prediction scores were generated for both test data and a portion of the training data, with results visualized in graphical form.

The process begins with uploading the dataset into the program as depicted in the methodology diagram. Once the data is loaded, the x-axis and y-axis in the graph show the number of records associated with each class label. By clicking the designated button, the program reads the dataset, normalizes it, replaces any missing values with zeros, and splits it into training and test sets.

The graph in Figure 2 displays the dataset distribution across class labels, helping to visualize the data balance.

The chart in Figure 3 compares the accuracy, recall, precision, and F-score of the seven algorithms. Notably, Random Forest and LightGBM achieved the highest scores across all metrics.

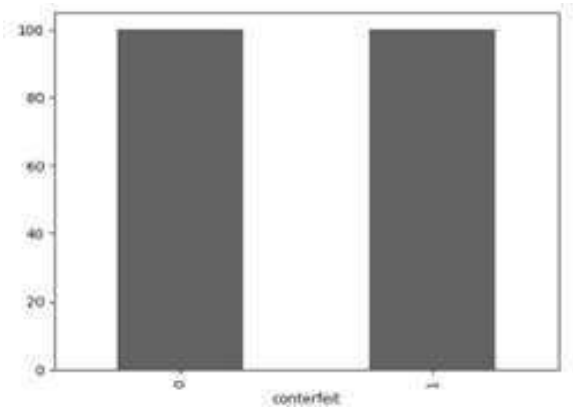

Figure 2. Counterfeit Detection Graph

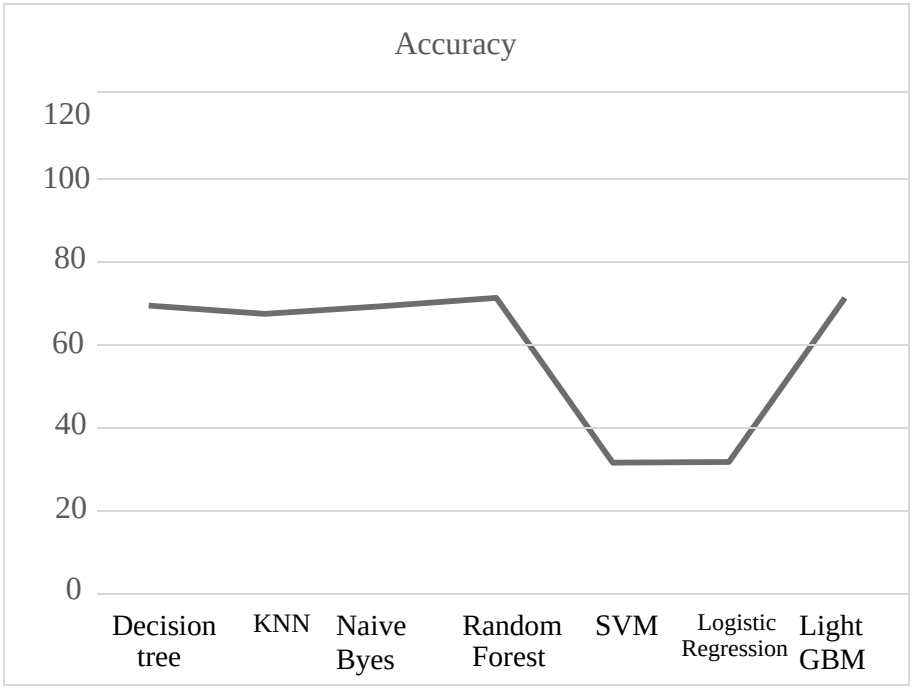

Figure 3. Accuracy Score Comparison

Table 1 below provides a summary of the accuracy, precision, recall, and F-score for each algorithm. Random Forest and LightGBM emerged as the top-performing algorithms, achieving perfect scores across all metrics, highlighting their effectiveness in counterfeit detection

Table 1. Performance Metrics Comparison

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
Decision Tree	97.5	97.7	97.3	97.5
KNN	95.0	95.6	94.7	94.9
Naive Bayes	97.4	97.8	97.3	97.4
Random Forest	100.0	100.0	100.0	100.0
SVM	47.5	23.7	50.0	32.3
Logistic Regression	47.8	23.9	50.0	32.2
LightGBM	100.0	100.0	100.0	100.0

Table 1 provides a detailed comparison of the performance of seven machine learning algorithms—Decision Tree, K-Nearest Neighbors (KNN), Naive Bayes, Random Forest, Support Vector Machine (SVM), Logistic Regression, and LightGBM—in terms of accuracy, precision, recall, and F-score. These metrics are critical in evaluating the model's effectiveness in distinguishing between real and counterfeit currency notes, providing insights into each algorithm's strengths and limitations.

Top Performers: Random Forest and LightGBM

Both Random Forest and LightGBM achieved perfect scores across all metrics, with an accuracy, precision, recall, and F-score of 100%. This indicates that these algorithms were highly effective in correctly classifying both real and counterfeit notes without any misclassifications.

The high performance of Random Forest can be attributed to its ensemble nature, which combines the predictions of multiple decision trees to improve overall accuracy and robustness. LightGBM, known for its efficiency and speed in handling large datasets, also demonstrates exceptional performance, highlighting its suitability for this type of classification problem.

Strong Performers: Decision Tree, KNN, and Naive Bayes

The Decision Tree, KNN, and Naive Bayes algorithms also demonstrated strong performance, achieving accuracy scores of 97.5%, 95.0%, and 97.4%, respectively.

The Decision Tree's high precision and recall values (97.7% and 97.3%) suggest that it was effective in correctly identifying both classes, although slightly less consistent than Random Forest due to the lack of ensemble averaging.

KNN, which relies on the similarity between data points, performs well with an accuracy of 95.0%, but it slightly lags in recall (94.7%), indicating some difficulty in capturing all true positives.

Naive Bayes, with scores close to those of Decision Tree, also shows strong performance, leveraging the assumption of feature independence to achieve high accuracy. However, Naive Bayes may be more prone to errors in cases where features are highly correlated.

Underperformers: SVM and Logistic Regression
Support Vector Machine (SVM) and Logistic Regression show considerably lower performance metrics across all categories. SVM achieved an accuracy of only 47.5%, with low precision (23.7%) and an F-score of 32.3%. Logistic Regression performed similarly, with an accuracy of 47.8% and similar low scores in precision and recall.

The poor performance of SVM and Logistic Regression may be attributed to the complexity of the dataset and the possible overlap between classes, making it difficult for linear classifiers to separate the classes effectively. These results indicate that SVM and Logistic Regression may not be suitable for this specific task of currency classification, especially when faced with intricate patterns that require more sophisticated decision boundaries.

Precision and Recall Insights
Precision and recall are particularly important metrics in counterfeit detection. High precision indicates that the algorithm is good at avoiding false positives (incorrectly classifying real notes as counterfeit), while high recall ensures that the model captures most of the counterfeit notes, minimizing false negatives.

Random Forest and LightGBM's perfect precision and recall scores reflect a balanced ability to avoid both types of errors, making them ideal choices for real-world applications where both types of misclassifications are costly.
Decision Tree and Naive Bayes also maintain high precision and recall, showing reliability in most cases. However, KNN's slightly lower recall indicates it may miss some counterfeit notes, which could be problematic depending on the application.

The analysis of Table 1 highlights that ensemble methods like Random Forest and gradient-boosting algorithms like LightGBM are exceptionally well-suited for the task of counterfeit detection due to their ability to capture complex patterns and dependencies within the data.

While Decision Tree, KNN, and Naive Bayes offer solid performance, they fall short of the top performers, which might affect reliability in high-stakes applications. SVM and Logistic Regression, however, demonstrated limitations that make them less ideal for counterfeit detection tasks, likely due to the nonlinearity and complexity of the dataset.

Overall, this analysis suggests that Random Forest and LightGBM should be prioritized for applications requiring high accuracy and reliability in counterfeit detection systems. This study also emphasizes the importance of choosing models that align with the data's characteristics to optimize classification performance.

Conclusion

In this study, we utilized the Banknote Authentication dataset from the University of California Irvine Machine Learning Repository to evaluate the performance of various machine learning algorithms, including Support Vector Machine, Logistic Regression, Naive Bayes, Decision Tree,

Random Forest, and K-Nearest Neighbor. These algorithms, implemented from the Scikit-Learn (SML) library, were tested on the dataset at three different train-test ratios to assess their effectiveness in distinguishing genuine from counterfeit banknotes. The dataset consisted of 1,372 records with four key features and one target variable, representing characteristics essential for currency authentication.

Our results demonstrate that Random Forest and LightGBM achieved the highest performance across multiple evaluation metrics, including accuracy, precision, recall, and F-score, indicating their suitability for this classification task. This study highlights the potential of machine learning in automated counterfeit detection and emphasizes the importance of selecting robust models that align well with data characteristics. Future work could focus on optimizing these models further and exploring additional features to enhance the system's reliability in real-world applications.

References

Aoba, M., Kikuchi, T., & Takefuji, Y. (2003). Euro banknote recognition system using a three-layered perceptron and RBF networks. *IPSJ Transactions on Mathematical Modeling and Its Applications*, May 2003. https://www.researchgate.net/publication/265046877_Euro_Banknote_Recognition_System_Using_a_Three-layered_Perceptron_and_RBF_Networks

Desai, S., Kabade, S., Bakshi, A., Gunjal, A., & Yeole, M. (2014). Implementation of multiple kernel support vector machine for automatic recognition and classification of counterfeit notes. *International Journal of Scientific & Engineering Research*, October 2014. https://api.semanticscholar.org/CorpusID:110087891

Gigliarano, C., Figini, S., & Muliere, P. (2014). Making classifier performance comparisons when ROC curves intersect. *Computational Statistics and Data Analysis, 77*, 300–312. https://doi.org/10.1016/j.csda.2014.03.008

Hassanpour, H., & Farahabadi, P. M. (2009). Using Hidden Markov Models for paper currency recognition. *Expert Systems with Applications, 36*(6), 10105–10111. https://doi.org/10.1016/j.eswa.2009.01.057

Huang, Z., Chen, H., Hsu, C., Chen, W., & Wu, S. (2003). Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems, 37*(4), 543–558. https://doi.org/10.1016/s0167-9236(03)00086-1

Kumar, C., & Dudyala, A. K. (2015). Bank note authentication using decision tree rules and machine learning techniques. *International Conference on Advances in Computer Engineering and Applications*. https://doi.org/10.1109/icacea.2015.7164721

Nastoulis, C., Leros, A., & Bardis, N. (2006). Banknote recognition based on probabilistic neural network models. *International Conference on Systems*, 795–798. http://www.wseas.us/e-library/conferences/2006cscc/papers/534-095.pdf

Omatu, S., Yoshioka, M., & Kosaka, T. (2007). *Bank note classification using neural networks*. https://www.semanticscholar.org/paper/Bank-note-classification-using-neural-networks-

Omatu-Yoshioka/bfe10fe2dd52a90acb3e8da47bd9c539c97d04a4

Prime, E. L., & Solomon, D. H. (2010). Australia's Plastic Banknotes: Fighting Counterfeit Currency. *Angewandte Chemie International Edition*, *49*(22), 3726–3736. https://doi.org/10.1002/anie.200904538

Roy, A., Halder, B., & Garain, U. (2010). Authentication of currency notes through printing technique verification. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)* (pp. 383–390). http://dx.doi.org/10.1145/1924559.1924610

Singh, R., & Agarwal, R. (2018). Currency recognition using image processing and machine learning techniques. *International Journal of Computer Applications, 179*(18), 31-37. https://doi.org/10.5120/ijca2018917616

Walia, E., & Pal, A. (2015). Paper currency recognition using artificial neural network and K-means clustering. *Procedia Computer Science*, *46*, 1048-1054. https://doi.org/10.1016/j.procs.2015.02.101