



The Detection of Counterfeit Banknotes Using Ensemble Learning Techniques of AdaBoost and Voting

Rihab Salah Khairy¹**Ameer Saleh Hussein¹****Haider TH. Salim ALRikabi²**¹*Directorate General of Education in Babylon, Babylon, Iraq*²*Electrical Engineering Department, College of Engineering, Wasit University, Wasit, Iraq*

* Corresponding author's Email: rehabsalah08@gmail.com

Abstract: The movement of cash flow transactions by either electronic channels or physically created openings for the influx of counterfeit banknotes in financial markets. Aided by global economic integration and expanding international trade, attention must be geared at robust techniques for the recognition and detection of counterfeit banknotes. This paper presents ensemble learning algorithms for banknotes detection. The AdaBoost and voting ensemble are deployed in combination with machine learning algorithms. Improved detection accuracies are produced by the ensemble methods. Simulation results certify that the ensemble models of AdaBoost and voting provided accuracies of up to 100% for counterfeit banknotes.

Keywords: Banknote detection, Ensemble methods, AdaBoost, Voting, Machine learning.

1. Introduction

The value placed on the currency by any country establishes the currency as great assets for economic growth. The financial institution's initiation of electronic currency in the move to reduce the circulation of physical notes has been labelled positive. However, banknotes are still very much in circulation, and this motivated the proliferation of counterfeit banknotes that leads to susceptibility and loss of profit among traders and banks. There is a higher value attached to banknotes than coins, thus increasing the susceptible to counterfeiting, and a higher economical risk [1, 2]. The counterfeit banknotes are built with security features that make it difficult to be detected. Thus, advanced recognition methods to see beyond the obfuscation integrated into the banknotes are imperative to classify the genuine from the counterfeit ones. There are different numbers of techniques that have been proposed and developed for tackling banknotes counterfeiting. They comprise of support vector machines, artificial neural networks, genetic algorithms, principal component analysis, gray-level co-occurrence matrix,

k-nearest neighbours, linear discrimination analysis, etc [3-8].

In this paper, an alternative method for the detection of banknotes is proposed based on ensemble techniques of Boosting and Voting algorithms used with a selection of ten (10) standard machine learning algorithms. It adopted ensemble learning algorithms of AdaBoost and Voting (with a combination rule of the average of probabilities). The integration of the machine learning algorithms into the ensemble methods results in a classification outcome for effective detection. Datasets from a well-known database were retrieved for experimentations and evaluated with standard metrics for a fair comparison.

The organization of the paper is as follows: Section 2 summarizes relevant literature concerning banknotes detection. Section 3 discusses the algorithms consisting of all the machine learning and ensemble algorithms used in this paper. Experimentations are analyzed in Section 4. Section 5 occupies the conclusion and future works.

2. Related works

There has been lots of research conducted for the detection of fake and counterfeit banknotes in literature. This section reviews the various work carried out to solve the problem of banknote detection. Sarfraz [9] proposed the application of a radial basis function (RBF) neural network for the recognition of banknotes currency. The focus was directed at Saudi Arabia's currency and after feature extraction and classification; the proposed model produced reasonable accuracy results. An improved feed-forward neural network is based on a mix-margin principle that requires limited samples for a better flexible network and accuracy [10]. The margin-based feed-forward neural network was applied to banknote data and achieved higher accuracy in comparison to ANN, SVM, and AdaBoost algorithms. To efficiently detect non-genuine banknotes, a Chebyshev multilayer perceptron neural network with levenberg Marquardt backpropagation is proposed [11]. It operates mainly on Chebyshev orthogonal polynomial which is used as functional expansion. The levenberg Marquardt is the learning part of the neural network. There was an improvement in performance from 55% to 90% as to other algorithms. A grid-based local outlier factor (LOF) algorithm was presented for adequate banknote outlier detection [12]. The grid methodology reduces the computational complexity of LOF by breaking down the data space into smaller subspaces. The LOF of individual subspaces is calculated thus significantly boosting the performance of LOF. In the detection of counterfeit banknotes, a combination of hardware and software architecture is used [13]. The architectural system uses Euro banknotes for recognition and authentication and made up of an infrared camera, a microprocessor, and a software component of smart algorithms. A glass is placed in the focal plane of the camera to obtain clear cut images. Thereafter, the algorithms take charge of processing the images for validation. The proposed method recognizes the forged banknotes. The authors in [14] developed a framework for self-training based on finding density peaks. The frameworks house two essential layers: the first layer is discovering the real structure of entire data space by searching and finding density peaks of data; while the second layer is the integration of the real structure of entire data space into the self-training process to iteratively train a classifier. Using the classifiers of SVM, k -NN, and classification and regression tree (CART), the proposed framework shows to surpass other methods for banknote authentication. According to Sargano *et*

al. [15], they claim to be the first to conduct recognition for Pakistani banknotes. Ten different currency papers totally 175 banknotes were used. Essential features are extracted from the notes, which are passed as input for training a three-level feed-forward back-propagation neural network. Accurate detection was achieved after experimentations. An empirical approach for currency identification through automation was presented by authors in [2]. The feature vectors are extracted from gray-level histogram shape descriptors and banknote image texture. Classification of the feature vectors is performed by the feed-forward neural network, and accuracy of 98.6% was generated in comparison to pattern recognition feed-forward neural network (PRFNN), cascade forward neural network (CNN), and AdaBoost. Kumar and Dudyala [16] carried out an extensive experimental investigation on banknote authentication using several machine learning algorithms. These learning algorithms include a probabilistic neural network (PNN), MLP, RBF, decision tree, and naïve Bayes. Results assert that the MLP and decision tree gave performances that classify the banknote data for significant predictions. Making use of the multispectral sensors developed for ATMs, multispectral images of banknotes are obtained for improved detection of counterfeit banknotes [17]. The images have low resolutions of 50 dots per inch with RGB images and infrared images. An accuracy of 100% for the counterfeit notes was recorded. A multi-kernel SVM was proposed for banknote detection where there are $m \times n$ partitions of the banknotes, and the luminance histograms of the partitions are selected for the algorithm's input [1]. With the individual partitions linked to its kernels, the combination of the multiple kernels into a matrix is carried out by a linearly weighted combination. Determining optimal weights is through semi-definite programming (SDP). It was found that there tends to be an increase in time for SDP. Thus, an attempt to resolve this involves assuming a non-negativity of the kernel weights and setting some of the weights to be unity. The proposed multi-kernel SVM outperforms compared algorithms. The extraction of relevant features for accuracy recognition of banknotes has become imperative. Gai *et al.* in [18] looked into this by proposing a new feature extraction procedure by tapping in the merits of the quaternion wavelet transform (QWT). One shift-invariant magnitude and three phases based on quaternion algebra are generated by QWT. The application of generalized Gaussian density was used to retrieve the statistical attributes of QWT. The backpropagation neural network classifies the final extracted features. A gene expression programming

ensemble was proposed for classifying banknotes data [19]. Several security features of the Indian banknotes for exhaustive analysis was the focus in [20]. 1000 banknote samples comprising of 500 each of genuine and fake notes are experimented on. The authors engaged the k-means, NN, and SVM for classification of notes. The colour and texture features for classifying banknotes came to the fold in the work of García-Lamont *et al.* [21]. With the colour feature modelled under RGB, texture feature on the other hand is modeled with binary patterns methods. For classification, the linear vector quantization (LVQ) networks are used and a non-parametric test based on G statistic is performed. High recognition accuracy rate is obtained by the LVQ classifier. A comprehensive review was conducted in the areas of the recognition and detection of banknotes [22]. The review targets four keys research components namely; banknote recognition, counterfeit banknote detection, serial number recognition, and fitness classification with the application of various sensors. In-depth discussions and analysis of the different recognition methods, feature extractions techniques, and modes of algorithmic classification were reported.

3. Machine learning and ensemble

This section encompasses the algorithms used for experimentations in this paper. A total of ten (10) algorithms are employed. Also, ensemble algorithms of AdaBoost and Voting are drafted for use. Below are detailed discussions of all the algorithms.

3.1 Naïve bayes

Dependent on $R(S_a = s_a / T = t_a)$, which is known as class conditional probabilities, naïve bayes [23] is used to understand the probabilities for individual feature S_a assigned to class label t_a . Engaging in Bayesian rule, classification is executed with the help of posterior probability as in Eq. (1).

$$\frac{R(T = t_1 | S_1 = s_1, S_2 = s_2, S_3 = s_3, \dots, S_n = s_n)}{R(S_1 = s_1, S_2 = s_2, S_3 = s_3, \dots, S_n = s_n)} = \frac{R(T = t_1)R(S_1 = s_1, S_2 = s_2, S_3 = s_3, \dots, S_n = s_n | T = t_1)}{R(S_1 = s_1, S_2 = s_2, S_3 = s_3, \dots, S_n = s_n)} \quad (1)$$

It is imperative to note that features are independent of one another. This is conditional and aligns with Eq. (2).

$$R(S_1 = s_1, S_2 = s_2, S_3 = s_3, \dots, S_n = s_n | T = t_i) = \prod_{a=1}^n R(S_a = s_a | T = t_a) \quad (2)$$

Eq. (2) above simplifies the approximation of class conditional probabilities [24, 25].

3.2 3.2. Random forest

Random forest (RF) is a non-parametric method to overcome the limitations posed by regression trees [26,27]. It is an ensemble technique for classification or regression tasks triggered through bootstrap samples of the training data, with the aid of random feature selection in the tree induction process. For a prediction to occur, aggregations of predictions either by majority vote for classification or averaging for regression are initiated for the ensemble. The RF tends to boost regression trees' performance by compromising interpretability. This means having building blocks of several trees $\hat{f}^1(x), \hat{f}^2(x), \hat{f}^3(x), \dots, \hat{f}^B(x)$ and thereafter attaining low variance regression model by averaging of predictions, depicted by:

$$\hat{f}_{avg}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x) \quad (3)$$

where B represents the number of trees. The growth of trees by RF is dependent on a subset m of ρ predictors. Thus, satisfying the rule of choosing $m \approx \sqrt{\rho}$ predictors.

3.3 k-Nearest neighbor

The description of algorithm k -nearest neighbor (k -NN) is the reliance on some specific connotations namely R_m, q , and V_t . R_m refers to a similarity measure that links every pair of data either real or integer. q accounts for total data cluster injected for classification. The data vector that the classifier employs is denoted by V_t [28]. The Euclidean distance is the most common distance metric deployed by k -NN to compute the correlation existing among points [29], as indicated in Eq. (4).

$$E(f, g) = \left(\sum_{b=1}^u (f_b - g_b)^2 \right)^{1/2} \quad (4)$$

where $E(f, g)$ denotes Euclidean distance, f , and g are data points.

3.4 Fuzzy nearest neighbor

The process of classifying a test object owing to the similarity concerning a specified K -nearest neighbor and their respective membership degrees is

ascribed to the proposition of the fuzzy K -nearest neighbor (FNN) algorithm [30], [31]. Algorithm 1 shows the pseudo-code of FNN. Given that an object y belongs to a class C , the similarity is formulated as:

$$C'(y) = \sum_{x \in N} R(x, y)C(x) \quad (5)$$

where N connotes the set of object y 's K -nearest neighbors. $R(x, y)$ is the similarity of x and y and lies between the value of $[0, 1]$. It can also be defined traditionally as:

$$R(x, y) = \frac{\|y - x\|^{-2/(m-1)}}{\sum_{j \in N} \|y - j\|^{-2/(m-1)}} \quad (6)$$

where $\|\cdot\|$ depicts Euclidean norm, and m is used for handling the similarity's weight.

3.5 Sequential minimal optimization

The goal of sequential minimal optimization (SMO) is to train the support vector machines (SVMs). Basically to dissolve associated SVM deficiencies in handling large-sized problems [32]. The concept of SVM goes thus; Concerning [33], if there exists a collection of data points $\{(H_\varepsilon, w_\varepsilon)\}_\varepsilon^p$; H_ε and p is input vector and all training data. The process involved in training SVM for classification is analogous to finding a solution to the following:

$$\begin{aligned} \text{maximize: } Q(\delta_\varepsilon) = & \\ \sum_{\varepsilon=1}^p \delta_\varepsilon - \frac{1}{2} \sum_{\varepsilon=1}^p \sum_{\mu=1}^p \delta_\varepsilon \delta_\mu w_\varepsilon w_\mu \kappa(H_\varepsilon, H_\mu) & \quad (7) \end{aligned}$$

$$\begin{aligned} \text{subject to: } \sum_{\varepsilon=1}^p \delta_\varepsilon w_\varepsilon = 0, & \quad (8) \\ 0 \leq \delta_\varepsilon \leq c, \quad \varepsilon = 1, \dots, p & \end{aligned}$$

where $\kappa(H_\varepsilon, H_\mu)$, δ_ε , and c connotes kernel function, Lagrange multiplier, user-determined regularization constant respectively. The widespread kernel function is the Gaussian function. If the problem in (7) becomes resolved, a unique data sequence is identified by decision function in (9) for the class label.

$$\text{function}(H) = \sum_{\varepsilon=1}^p \delta_\varepsilon w_\varepsilon \kappa(H_\varepsilon, H) + b \quad (9)$$

with b acquired from Eq. (7).

The SVM fails to deal with QP problems of large sizes. In resolving this, the SMO disintegrate enormous QP task into sub-problems. Optimization of a training data sequence subset in each phase, which is called a working set. Two working sets are used to mitigate the QP sub-problems with a simple systematic technique [34]. A set of rules are vital in specifying two δ_ε . SMO adjusts quadratically the total data sequence.

3.6 Logistic regression

Logistic regression (LR) is a statistical method used to estimate the probability of occurrence of a binary outcome based on a set of explanatory features. It allows describing the influence of the considered factors on the analysed dichotomous variable. On the other hand, if the dependent variable has at least 3 unordered categories, then multinomial logistic regression (MLR) should be used. The MLR method was created on the basis of the concepts of binomial logistic regression and has the same basic setup, thus it can be said that it is an extension of logistic regression [35-37].

In the work done by Le Cessie and Van Houwelingen [38], a ridge values of 1×10^8 was suggested for the log likelihood calculation. There exist modifications to for the classification purpose [39]. If there are k classes for n instances with m attributes, the parameter matrix B to be calculated will be a $m * (k - 1)$ matrix. The probability for class j with the exception of the class is as in Eq. (10).

$$P_j(X_i) = \frac{\exp(X_i B_j)}{\sum_{j=1}^{k-1} \exp(X_i B_j) + 1} \quad (10)$$

The last class has probability as shown in Eq. (11).

$$1 - \sum_{j=1}^{k-1} P_j(X_i) = \frac{1}{\sum_{j=1}^{k-1} \exp(X_i B_j) + 1} \quad (11)$$

Therefore, the negative multinomial log-likelihood is represented as follows:

$$\begin{aligned} L = - \sum_{i=1}^n \{ \sum_{j=1}^{k-1} Y_{ij} \times \ln(P_j(X_i)) + (1 - \sum_{j=1}^{k-1} Y_{ij}) \times \ln(1 - \sum_{j=1}^{k-1} P_j(X_i)) \} + \text{ridge} \times (B^2) & \quad (12) \end{aligned}$$

To find the matrix B for which L is minimized, a Quasi-Newton Method is used to search for the optimized values of the $m*(k-1)$ variables. It should be mentioned that before using the optimization procedure, the matrix B is squeezed into a $m*(k-1)$ vector [38,39].

3.7 Multilayer perceptron

The MLP is known as the most widely used artificial neural networks (ANNs) [40]. ANNs nonlinear generically approximating algorithms with major pattern recognition and classification applications [41,42]. The MLP consists of two layers, i.e., input and output layers, formed by simple nerve cells, and also contains solitary or diverse layers between input and output layers. Perceptrons are simply called nerve cells. The perceptron produces a single output from several inputs by linear fusion based on its input weights. A nonlinear transfer function determines the output. It is shown in Eq. (13) (Singh *et al.*, 2012)[43].

$$y = f(\sum_{i=1}^n w_i x_i + b) \quad (13)$$

Where w_i , x_i , b , f , and y represent the weight vector, input vector, bias, transfer function, and output respectively. Denoted in Eq. (14) is the logistic sigmoid transfer function defined as:

$$\frac{1}{(1+e^{-x})} \quad (14)$$

3.8 Decision stump

A decision stump (DS) is an algorithm of a one-tier decision tree with only an internal node (root) linked directly with terminal nodes (leaves) [44]. The DS forecasts can be based on a single input value and can be referred a one-rule. A DS is formulated as:

$$f(x; j, b, s) = s \cdot \text{sign}(x_j - b) \quad (15)$$

where s is in the range of $\{-1, 1\}$. The parameters j , b , and s describe a decision stump. The DS is therefore a shifted step function for fixed values of s and b , for which \mathbf{x} can be allocated, a label solely centered on the j^{th} predictor \mathbf{x}_j .

3.9 Random tree

Having a set of possible trees, a random tree [45] with each node of n random features, is drawn randomly. This means there is an equal opportunity for individual tree to be sampled. This randomness is

uniformly distributed and can be produced efficiently. Accurate predictions of the algorithm are acquired through large set of random trees.

3.10 JRip

The JRip [46,47] is an implementation of a propositional rule learner called Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [48]. It is a rule induction algorithm that takes advantage of global optimization step in generating rule sets with consideration of both quality and length of rules. Deploying a sequential covering strategy, set of rules is created for each class value. Reconsideration of each rule takes effect and different sequences of rules are produced by reduced-error pruning. If one of the new rule sequences has smaller length, it displaces the rule.

3.11 Boosting (AdaBoost) ensemble algorithm

The AdaBoost algorithm has become the most widely and effectively used boosting technique. It is a simplification derived from Adaptive Boosting. The AdaBoost ensemble is constructed by combining weak learners [44]. Weights are attached to the training samples and adjustments are made on the weights after creating each weak learner. Low weights are assigned to easy samples that are classified correctly by the weak learner, while high weights are for the hard or misclassified samples. By maintaining additional weights, AdaBoost concentrates on more difficult samples. Weak learners are called by AdaBoost by applying the following steps.

- 1) **Input:** This comprise of a set training samples with labels $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, the weak learner (e.g., lets us use k -NN), and number of cycles T .
- 2) **Initialization:** The weights of training samples are initialized: $w_i^{(1)} = 1/N, i = 1, 2, \dots, N$.
- 3) **Loop:** This is repeated for $t = 1$ to T
 - a. The weak learning algorithm is used for the training of the weighted samples of a weak classifier $h^{(t)}$.
 - b. The training error of $h^{(t)}$ is calculated as: $\epsilon^{(t)} = \sum_{i=1}^N w_i^{(t)}, y_i \neq h^{(t)}(\mathbf{x}_i)$.
 - c. The setting of the weight for weak classifier $h^{(t)}$: $\alpha^{(t)} = \frac{1}{2} \ln \left[\frac{1-\epsilon^{(t)}}{\epsilon^{(t)}} \right]$.
 - d. Updating the weights of training samples is performed using: $w_i^{(t+1)} = w_i^{(t)} \frac{\exp[-\alpha^{(t)} \cdot y_i h^{(t)}(\mathbf{x}_i)]}{c^{(t)}}$

, $i = 1, 2, \dots, N$, where $C^{(t)}$ is a normalization constant, and $\sum_{i=1}^N w_i^{(t+1)} = 1$.

4) **Output:** This is translated as $f(\mathbf{x}) = \text{sign}\left[\sum_{t=1}^T (\alpha^{(t)} \cdot h^{(t)}(\mathbf{x}))\right]$.

3.12 Voting ensemble algorithm

The voting classifier algorithm is a simple but very effective ensemble algorithm. The algorithm capitalizes on the defect of one algorithm which could be an advantage to another classifier. Voting takes the predictions of classifiers and combines them into a final prediction output [49].

Let the number N of selected classifiers be represented by S_1, \dots, S_N and $\mathfrak{R} = \{S_i : i = 1, 2, 3, \dots, N\}$. If there exists M number of output classes, the voting classifier ensemble takes the following steps [50]: Determine how each classifier S_i will be combined by votes V for optimizing a function $F(V)$. The representation of V is a real array of sizes $N \times M$. $V(i, j)$ denotes the weight of the vote of the i^{th} classifier with j^{th} class. The more confident the

classifier is, the higher the weight assigned, while lesser weight is designated for the less confident classifier. $V(i, j) \in [0, 1]$ is the degree of confidence of i^{th} classifier for the j^{th} class. With reliance on a combination rule, the weights play a major role during combining prediction output of classifiers.

4. Experimental setup and results

The banknotes datasets used for experimentations are provided and retrieved from UCI Machine Learning Repository through <http://archive.ics.uci.edu/ml> [51] and as recorded in the work of Flury and Riedwyl [52]. The datasets are Banknote authentication and Swiss Franc banknotes data. The Banknote authentication dataset was taken from images of genuine and forged banknote-like specimens. The objective is to distinguish and discover forged banknotes from authentic ones [53]. It consists of three characteristic features. In conjunction with the class variable, a fourth feature called image entropy captures the dataset, along with the class variable, embodies the dataset. There is a non-existence of missing values among 1372 instances. With respect to the Swiss Franc bank notes

Table 1. Results of Banknote Authentication dataset for Individual Algorithms

Algorithms	Accuracy (%)	DR (%) for Genuine	DR (%) for Forged	MCC (%)
Naïve Bayes	84.30	88.20	79.30	68.00
Random Forest	99.30	99.30	99.30	98.70
k-NN	99.90	99.70	100	99.70
FNN	99.80	99.60	100	99.60
SMO	98.00	96.50	100	96.10
LR	99.10	99.20	99.00	98.20
MLP	99.90	100	99.80	99.90
Decision Stumps	84.30	80.80	88.70	69.10
Random Tree	98.80	98.60	99.00	97.50
JRip	98.50	99.20	97.70	97.10

data, it consists of 200 instances of old Swiss 1000-Franc bank notes divided into 100 instances of genuine notes and 100 instances of counterfeit notes. It describes the characteristics of the bank note with 6 input variables and a class variable. Both datasets are trained and tested with ten (10) standard machine learning algorithms namely; naïve bayes, random forest, k -nearest neighbor, fuzzy nearest neighbor, sequential minimal optimization, logistic regression, multilayer perceptron, decision stumps, random tree, and JRip together with their corresponding usage with Boosting ensemble and Voting ensemble algorithms. The Boosting algorithm of AdaBoostM1 and Voting algorithm (with combination rule of average of probabilities) are used. The Waikato environment for knowledge analysis (WEKA) takes the centre stage for running all the experiments. A 10-fold cross-validation is used training and evaluating. The process entails dividing the dataset into ten subsets of equal size with nine subsets for the training data, and one subset as the test data. An average mean of each results is collated.

4.1 Assessment measures

The performance metrics to evaluating algorithms' effectiveness are the accuracy (ACC), detection rate (DR) (true positive rate), and Matthews correlation coefficient (MCC). The terms are defined as follows and depicted in Eq. (16) through to Eq. (18):

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (16)$$

$$DR = \frac{TP}{TP+FN} \quad (17)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (18)$$

where TP and FP are the true positives and false positives, while FN and TN are the false negatives and true negatives.

4.2 Simulation results

The simulations are conducted using WEKA on 3.40GHz Intel® Core i7 Processor with 4GB of RAM. The results after a series of experiments for each dataset are tabulated and graphed. Table 1 accommodates the results of standard algorithms used with Banknote authentication data. The accuracy rates are very high between 98% to above 99% except naïve bayes and decision stumps

producing 84.30% each. The highest accuracy of 99.90% is attributed to k -NN and MLP. Rates for genuine and forged notes vary with interesting values. Algorithms with a detection rate of 100% are given by MLP for genuine notes and k -NN, FNN, and SMO for forged notes. Naïve bayes produce the lowest MCC of 68%, while the highest MCC is from MLP at 99.90%.

The AdaBoost ensemble algorithm has been applied to all the standard algorithms. Their results are in Table 2. AdaBoost improved on the accuracy performance of the algorithms while also produced similar result outcomes as with some of the algorithms. There is an increment in performance values for naïve bayes, random forest, SMO, decision stumps, and JRip in terms of accuracy, detection rates, and MCC. Only the random tree had a decline in percentage values. To extract the best results for the voting ensemble, a combination of the algorithms was performed. Afterward, a total of nine (9) combinations of algorithms are reported in Table 3. The accuracy rate yielded above 99%, except for RT + RF at 98.80%. The highest accuracy of 100% is achieved by k -NN + LR, FNN + LR, and MLP + FNN respectively. Detection rates for genuine notes have similar highest rates as accuracy, and forged notes hit a high at 100% for all algorithms except that of SMO + RT, LR + MLP, and RT + RF with 99%, 99.70%, and 99% rates accordingly.

For fair performance comparison, the results that are reported by Wu et al. [14], Kumar and Dudyal [16], and Jędrzejowicz and Jędrzejowicz [19] are selected. In their works, the same banknote authentication dataset has been used with 10-fold cross-validation as employed in our work. The results are shown in Table 4. For Wu et al. [14], a framework based on self-training semi-supervised classification (SSC) that integrates the algorithms of SVM, k -NN, and CART produces 94.50%, 85.70%, and 67% accuracy respectively. The MLP and probabilistic neural network (PNN) accounted for the best accuracy rates at 98.83% and 98.60% for the work by Kumar and Dudyal [16]. Also, a GEP-Ensemble method gave banknote detection accuracy of 96.82% [19]. As compared, the best accuracy results from experiments in our paper generated above 99% for most of the individual algorithms as well as when used with the AdaBoost ensemble. Very fascinating results were returned for the Swiss Franc banknotes dataset as imputed for all algorithms in Table 5. The algorithms registered similar percentage values with 98.50% occurring frequently for accuracy, 99% and 98% are equally shared for detection rate on genuine

Table 2. Results of banknote authentication for boosting (AdaBoost) algorithms

Algorithms	Accuracy (%)	DR (%) for Genuine	DR (%) for Forged	MCC (%)
Naïve Bayes	98.30	97.50	99.30	96.60
Random Forest	99.60	99.50	99.80	99.30
<i>k</i> -NN	99.90	99.70	100	99.70
FNN	99.80	99.70	99.80	99.60
SMO	98.90	100	99.80	99.90
LR	99.10	99.20	99.00	98.20
MLP	99.90	100	99.80	99.90
Decision Stumps	94.50	94.90	93.90	88.80
Random Tree	98.30	98.30	98.20	96.50
JRip	99.60	99.60	99.50	99.10

Table 3. Results of Banknote Authentication for Voting Ensemble

Algorithms	Accuracy (%)	DR (%) for Genuine	DR (%) for Forged	MCC (%)
NB + <i>k</i> -NN	99.90	99.70	100	99.70
RF + FNN	99.90	99.90	100	99.90
<i>k</i> -NN + LR	100	100	100	100
FNN + LR	100	100	100	100
SMO + RT	99.50	99.90	99.00	99.00
LR + MLP	99.80	99.90	99.70	99.60
MLP + FNN	100	100	100	100
DS + <i>k</i> -NN	99.90	99.70	100	99.70
RT + RF	98.80	98.70	99.00	97.60

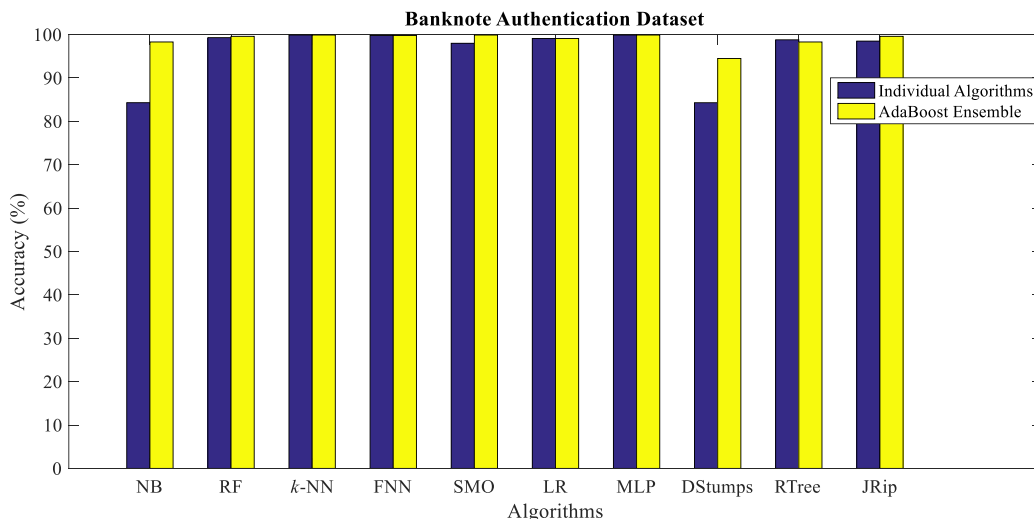


Figure. 1 The accuracy rates of banknote authentication for individual algorithms and AdaBoost

Table 4. Results of compared algorithms for banknote authentication data

Work Referenced	Methods	Accuracy (%)
Wu et al. [14]	Self Labeled SSCD _{peaks} + SVM	94.50
	Self Labeled SSCD _{peaks} + k-NN	85.70
	Self Labeled SSCD _{peaks} + CART	67.00
Kumar and Dudyala [16]	MLP	98.83
	PNN	98.60
Jędrzejowicz and Jędrzejowicz [19]	GEP-Ensemble	96.82

notes, 99% appeared most for counterfeit notes but 100% detection rates showed to be the highest rate. As regards MCC, 97% occurred more times for all algorithms. The results from AdaBoost on all algorithms are tabulated in Table 6. Accuracy rates range between 96.50% for the random tree to 99.50% for random forest and FNN. The genuine fraud rates of naïve bayes, random forest, FNN, SMO, LR, decision stumps, and JRip are at 99%, while others are assigned 98%. The best MCC rates are achieved by FNN and random forest at 99%. The lowest MCC of 93% is produced by random tree. Detection rates for counterfeit notes achieved 100% for random forest and FNN.

The voting ensemble results are in Table 7. Similar outcomes are reported to that of the AdaBoost ensemble. Superior accuracy at 99.50% is yielded for

RF + FNN. Most of the algorithms have detection rates of above 99% for genuine notes except with NB + k-NN and MLP + FNN at 98%. Perfect detection rates for counterfeit notes are recorded for NB + k-NN and RF + FNN. For MCC, six of the models gave 97%, and two models have 98% and RF + FNN superior value of 99%.

The voting ensemble results are in Table 7. Similar outcomes are reported to that of the AdaBoost ensemble. Superior accuracy at 99.50% is yielded for RF + FNN. Most of the algorithms have detection rates of above 99% for genuine notes except with NB + k-NN and MLP + FNN at 98%. Perfect detection rates for counterfeit notes are recorded for NB + k-NN and RF + FNN. For MCC, six of the models gave 97%, and two models have 98% and RF + FNN superior value of 99%.

Table 5. Results of swiss franc banknotes for individual algorithms

Algorithms	Accuracy (%)	DR (%) for Genuine	DR (%) for Forged	MCC (%)
Naïve Bayes	99.50	99.00	100	99.00
Random Forest	98.50	98.00	99.00	97.00
<i>k</i> -NN	98.50	98.00	99.00	97.00
FNN	99.50	99.00	100	99.00
SMO	99.50	99.00	100	99.00
LR	98.50	99.00	98.00	97.00
MLP	98.50	98.00	99.00	97.00
Decision Stumps	98.50	98.00	99.00	97.00
Random Tree	98.50	99.00	98.00	97.00
JRip	98.50	98.00	99.00	97.00

To have a fair comparison of the Swiss Franc banknote data, the work done by Jobe and Pokojovy [54] is used as shown in Table 8. While the proposed method in [54] produced an accuracy rate of 98%, the Adaboost and Voting method in this paper gave a high rate of 99.50% and 99.90% respectively.

4.3 Justification of proposed techniques compared to other methods

The proposed ensemble methods of Adaboost and Voting comes with great benefits. Their usage for effectiveness in the analysis are numerous. For Adaboost, the algorithm is execution procedure is fast, and it comes in handy as it is simple and easy to implement. Its flexibility of being capable to combine with any machine learning algorithms is exceptional. Furthermore, the Adaboost is less susceptible to overfitting and very versatile for usage beyond binary classification.

With respect to Voting, some form of bias by a single model is handled effectively by voting. Thus it operates a non-bias system. It is also possible that the Voting produces a better overall score than the best of the base estimators, as it aggregates the predictions of

multiple models and tries to cover for potential weaknesses of the individual models.

5. Conclusion and future works

In this paper, a proposition of boosting and voting ensemble model for banknotes detection are described and presented. The boosting ensemble of AdaBoost is used to enhance the performance efficiency of ten machine learning algorithms of naïve bayes, random forest, *k*-nearest neighbor, fuzzy nearest neighbor, sequential minimal optimization, logistic regression, multilayer perceptron, decision stumps, random tree, and JRip. The voting ensemble samples combinations of the ten algorithms nine different pairings resulting in very good accuracy rates. The experimental results on banknote authentication and Swiss franc bank note datasets showed that the ensemble algorithmic models are capable of improving the accuracy detection of individual algorithms. AdaBoost and voting account for a highest of 99.90% and 100% respectively with banknote authentication dataset, while 99.50% with Swiss franc dataset. Therefore, the proposed models

Table 6. Results of swiss franc banknotes for boosting (AdaBoost) algorithms

Algorithms	Accuracy (%)	DR (%) for Genuine	DR (%) for Forged	MCC (%)
Naïve Bayes	98.50	99.00	98.00	97.00
Random Forest	99.50	99.00	100	99.00
<i>k</i> -NN	98.50	98.00	99.00	97.00
FNN	99.50	99.00	100	99.00
SMO	99.00	99.00	99.00	98.00
LR	98.50	99.00	98.00	97.00
MLP	98.50	98.00	99.00	97.00
Decision Stumps	99.00	99.00	99.00	98.00
Random Tree	96.50	96.00	97.00	93.00
JRip	99.00	99.00	99.00	98.00

Table 7. Results of swiss franc banknotes for voting ensemble

Algorithms	Accuracy (%)	DR (%) for Genuine	DR (%) for Forged	MCC (%)
NB + <i>k</i> -NN	99.90	98.00	100	98.00
RF + FNN	99.50	99.00	100	99.00
<i>k</i> -NN + LR	99.00	99.00	99.00	98.00
FNN + LR	98.50	99.00	98.00	97.00
SMO + RT	98.50	99.90	98.00	97.00
LR + MLP	98.50	99.00	98.00	97.00
MLP + FNN	98.50	98.00	99.00	97.00
DS + <i>k</i> -NN	98.50	99.00	98.00	97.00
RT + RF	98.50	99.00	98.00	97.00

through experimentation and analysis confirm that it is very suitable and proficient for the detection of counterfeit banknotes. Future works can be directed towards expanding the algorithms for ensembling in

getting better classification results. Also, other techniques that are used in developing ensemble models aside from boosting and voting should be considered for banknotes detection.

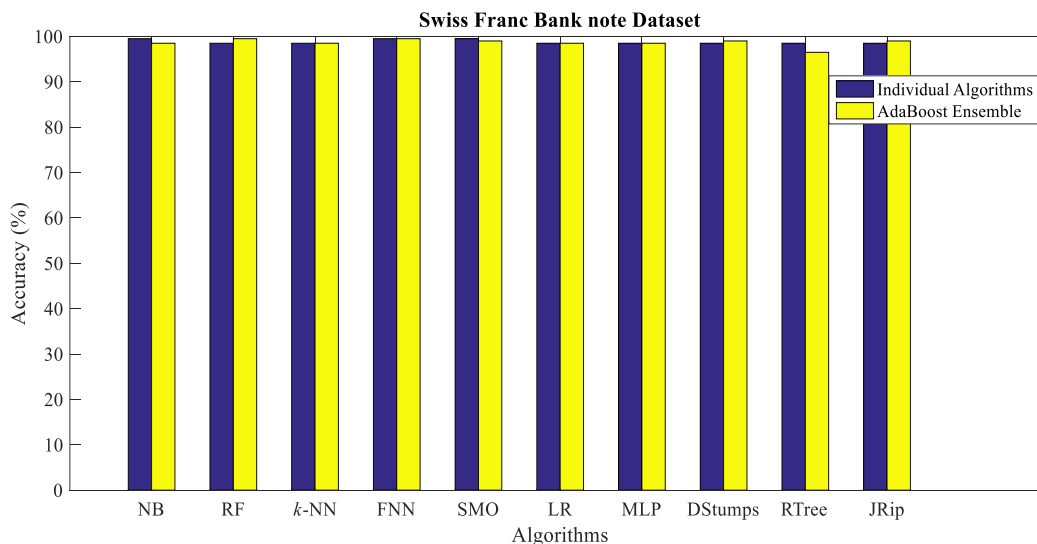


Figure. 2 The accuracy rates of swiss franc for individual algorithms and AdaBoost

Table 8. Results of compared algorithms for swiss franc banknote data

Work Referenced	Methods	Accuracy (%)
Jobe and Pokojovy [54]	FSRMCD-MAC	98.00

References

[1] C.-Y. Yeh, W.-P. Su, and S.-J. Lee, “Employing multiple-kernel support vector machines for counterfeit banknote recognition”, *Appl. Soft Comput.*, Vol. 11, No. 1, pp. 1439–1447, 2011.

[2] W. Q. Yan, J. Chambers, and A. Garhwal, “An empirical approach for currency identification”, *Multimed. Tools Appl.*, Vol. 74, No. 13, pp. 4723–4733, 2015.

[3] Y. Park, S. Kwon, T. Pham, K. Park, D. Jeong, and S. Yoon, “A high performance banknote recognition system based on a one-dimensional visible light line sensor”, *Sensors*, Vol. 15, No. 6, pp. 14093–14115, 2015.

[4] S. Shyju and A. Thamizharasi, “Indian currency identification using image processing”, *Int. J. Adv. Eng. Manag. Sci.*, Vol. 2, No. 5, 2016.

[5] S. A. Mousavi, M. Meghdadi, Z. Hanifeloo, P. Sumari, and M. R. M. Arshad, “Old and worn banknote detection using sparse representation and neural networks”, *Indian J. Sci. Technol.*, Vol. 8, pp. 913–918, 2015.

[6] F. Takeda, T. Nishikage, and S. Omatu, “Banknote recognition by means of optimized masks, neural networks and genetic algorithms”, *Eng. Appl. Artif. Intell.*, Vol. 12, No. 2, pp. 175–184, 1999.

[7] N. S. Mohamad, B. Hussin, A. S. Shibghatullah, and A. S. H. Basari, “Banknote authentication using artificial neural network”, In: *Proc. of the International Symposium on Research in Innovation and Sustainability*, Malacca, Malaysia, pp. 15–16, 2014.

[8] K. N. N. Hlaing, “First order statistics and GLCM based feature extraction for recognition of Myanmar paper currency”, In: *Proc. of the 31st IIER International Conf.* Bangkok, Thailand, 2015, pp. 1–6.

[9] M. Sarfraz, “An intelligent paper currency recognition system”, *Procedia Comput. Sci.*, Vol. 65, pp. 538–545, 2015.

[10] H. Xiao and X. Zhu, “Margin-Based Feed-Forward Neural Network Classifiers”, *arXiv Prepr.* arXiv1506.03626, 2015.

[11] U. Iqbal and R. Ghazali, “Chebyshev multilayer perceptron neural network with Levenberg Marquardt-back propagation learning for classification tasks”, In: *Proc. of International Conf. on Soft Computing and Data Mining*, pp. 162–170, 2016.

[12] J. Lee and N.-W. Cho, “Fast outlier detection using a grid-based algorithm”, *PLoS One*, Vol. 11, No. 11, p. e0165972, 2016.

- [13] A. Bruna, G. Farinella, G. Guarnera, and S. Battiato, "Forgery detection and value identification of Euro banknotes", *Sensors*, Vol. 13, No. 2, pp. 2515–2529, 2013.
- [14] D. Wu, M. Shang, X. Luo, J. Xu, H. Yan, W. Deng, and G. Wang, "Self-training semi-supervised classification based on density peaks of data", *Neurocomputing*, Vol. 275, pp. 180–191, 2018.
- [15] A. B. Sargano, M. Sarfraz, and N. Haq, "An intelligent system for paper currency recognition with robust features", *J. Intell. Fuzzy Syst.*, Vol. 27, No. 4, pp. 1905–1913, 2014.
- [16] C. Kumar and A. K. Dudyala, "Bank note authentication using decision tree rules and machine learning techniques", In: *Proc. of 2015 International Conf. on Advances in Computer Engineering and Applications*, pp. 310–314, 2015.
- [17] S. Baek, E. Choi, Y. Baek, and C. Lee, "Detection of counterfeit banknotes using multispectral images", *Digit. Signal Process.*, Vol. 78, pp. 294–304, 2018.
- [18] S. Gai, G. Yang, and M. Wan, "Employing quaternion wavelet transform for banknote classification", *Neurocomputing*, Vol. 118, pp. 171–178, 2013.
- [19] J. Jędrzejowicz and P. Jędrzejowicz, "Gene expression programming ensemble for classifying big datasets", In: *Proc. of International Conf. on Computational Collective Intelligence*, pp. 3–12, 2017.
- [20] A. Roy, B. Halder, U. Garain, and D. S. Doermann, "Machine-assisted authentication of paper currency: an experiment on Indian banknotes", *Int. J. Doc. Anal. Recognit.*, Vol. 18, No. 3, pp. 271–285, 2015.
- [21] F. Garcia-Lamont, J. Cervantes, and A. López, "Recognition of Mexican banknotes via their color and texture features", *Expert Syst. Appl.*, Vol. 39, No. 10, pp. 9651–9660, 2012.
- [22] J. Lee, H. Hong, K. Kim, and K. Park, "A survey on banknote recognition methods by various sensors", *Sensors*, Vol. 17, No. 2, p. 313, 2017.
- [23] P. Bermejo, J. A. Gámez, and J. M. Puerta, "Speeding up incremental wrapper feature subset selection with Naive Bayes classifier", *Knowledge-Based Syst.*, Vol. 55, pp. 140–147, 2014.
- [24] D. M. Farid, L. Zhang, C. M. RahM. A. Hossain, and R. Strachan, "Hybrid decision tree and naive Bayes classifiers for multi-class classification tasks", *Expert Syst. Appl.*, Vol. 41, No. 4, pp. 1937–1946, 2014.
- [25] A. Dangi and S. Srivastava, "Educational data classification using selective Naive Bayes for quota categorization", In: *Proc. of 2014 IEEE International Conf. on MOOC, Innovation and Technology in Education (MITE)*, 2014, pp. 118–121.
- [26] L. Breiman, "Random forests", *Mach. Learn.*, Vol. 45, No. 1, pp. 5–32, 2001.
- [27] H. A. Owolabi, M. Bilal, L. O. Oyedele, H. A. Alaka, S. O. Ajayi, and O. O. Akinade, "Predicting completion risk in PPP projects using big data analytics", *IEEE Trans. Eng. Manag.*, 2018.
- [28] H. Sahin and A. Subasi, "Classification of the cardiocogram data for anticipation of fetal risks using machine learning techniques", *Appl. Soft Comput.*, Vol. 33, pp. 231–238, 2015.
- [29] D. T. Larose and C. D. Larose, *Discovering knowledge in data: an introduction to data mining*, John Wiley & Sons, 2014.
- [30] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm", *IEEE Trans. Syst. Man. Cybern.*, No. 4, pp. 580–585, 1985.
- [31] D. T. Bui, Q. P. Nguyen, N.-D. Hoang, and H. Klempe, "A novel fuzzy K-nearest neighbor inference model with differential evolution for spatial prediction of rainfall-induced shallow landslides in a tropical hilly area using GIS", *Landslides*, Vol. 14, No. 1, pp. 1–17, 2017.
- [32] J. Platt and others, "Sequential minimal optimization: A fast algorithm for training support vector machines", *Technical Report MSR-TR-98-14*, 1998.
- [33] L. J. Cao, S. S. Keerthi, C. J. Ong, J. Q. Zhang, U. Periyathamby, X. J. Fu, and H. P. Lee, "Parallel sequential minimal optimization for the training of support vector machines", *IEEE Trans. Neural Networks*, Vol. 17, No. 4, pp. 1039–1049, 2006.
- [34] A. Barbero and J. R. Dorronsoro, "Momentum sequential minimal optimization: An accelerated method for support vector machine training", In: *Proc. of the 2011 International Joint Conf. on Neural Networks*, pp. 370–377, 2011.
- [35] D. W. Hosmer and S. Lemeshow, *Applied logistic regression*, Wiley New York, 2000.
- [36] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, Vol. 398. John Wiley & Sons, 2013.
- [37] A. J. Milewska, D. Jankowska, T. Wikesak, B. Acacio, and R. Milewski, "The application of multinomial logistic regression models for the assessment of parameters of oocytes and embryos quality in predicting pregnancy and

- miscarriage”, *Stud. Logic, Gramm. Rhetor.*, Vol. 51, No. 1, pp. 7–18, 2017.
- [38] S. Le Cessie and J. C. Van Houwelingen, “Ridge estimators in logistic regression”, *J. R. Stat. Soc. Ser. C (Applied Stat.)*, Vol. 41, No. 1, pp. 191–201, 1992.
- [39] D. H. Pandya, S. H. Upadhyay, and S. P. Harsha, “Fault diagnosis of rolling element bearing by using multinomial logistic regression and wavelet packet transform”, *Soft Comput.*, Vol. 18, No. 2, pp. 255–266, 2014.
- [40] B. T. Pham, D. T. Bui, H. R. Pourghasemi, P. Indra, and M. B. Dholakia, “Landslide susceptibility assessment in the Uttarakhand area (India) using GIS: a comparison study of prediction capability of naive bayes, multilayer perceptron neural networks, and functional trees methods”, *Theor. Appl. Climatol.*, Vol. 128, No. 1–2, pp. 255–273, 2017.
- [41] M. Zare, H. R. Pourghasemi, M. Vafakhah, and B. Pradhan, “Landslide susceptibility mapping at Vaz Watershed (Iran) using an artificial neural network model: a comparison between multilayer perceptron (MLP) and radial basic function (RBF) algorithms”, *Arab. J. Geosci.*, Vol. 6, No. 8, pp. 2873–2888, 2013.
- [42] A. Lasisi, R. Ghazali, M. M. Deris, T. Herawan, and F. Lasisi, “Extracting Information in Agricultural Data Using Fuzzy-Rough Sets Hybridization and Clonal Selection Theory Inspired Algorithms”, *Int. J. Pattern Recognit. Artif. Intell.*, Vol. 30, No. 9, p. 1660008, 2016.
- [43] A. Singh, M. Imtiyaz, R. K. Isaac, and D. M. Denis, “Comparison of soil and water assessment tool (SWAT) and multilayer perceptron (MLP) artificial neural network for predicting sediment yield in the Nagwa agricultural watershed in Jharkhand, India”, *Agric. Water Manag.*, Vol. 104, pp. 113–120, 2012.
- [44] H. Chen, Z. Lin, L. Mo, and C. Tan, “Identification of colorectal cancer using near-infrared spectroscopy and adaboost with decision stump”, *Anal. Lett.*, Vol. 50, No. 16, pp. 2608–2618, 2017.
- [45] Y. Zhao and Y. Zhang, “Comparison of decision tree methods for finding active objects”, *Adv. Sp. Res.*, Vol. 41, No. 12, pp. 1955–1959, 2008.
- [46] F. E. B. Otero, A. A. Freitas, and C. G. Johnson, “A new sequential covering strategy for inducing classification rules with ant colony algorithms”, *IEEE Trans. Evol. Comput.*, Vol. 17, No. 1, pp. 64–76, 2013.
- [47] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, 2016.
- [48] W. W. Cohen, “Fast effective rule induction”, In *Machine Learning Proc. 1995*, Elsevier, 1995, pp. 115–123.
- [49] U. K. Kumar, M. B. S. Nikhil, and K. Sumangali, “Prediction of breast cancer using voting classifier technique”, In: *Proc. of 2017 IEEE International Conf. on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, 2017, pp. 108–114.
- [50] S. Saha and A. Ekbal, “Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition”, *Data Knowl. Eng.*, Vol. 85, pp. 15–39, 2013.
- [51] K. Bache and M. Lichman, “UCI machine learning repository”, URL <http://archive.ics.uci.edu/ml>, Vol. 901, 2013.
- [52] B. Flury and H. Riedwyl, *Multivariate statistics: a practical approach*, Chapman and Hall, 1988.
- [53] V. Lohweg, J. L. Hoffmann, H. Dörksen, R. Hildebrand, E. Gillich, J. Hofmann, and J. Schaede, “Banknote authentication with mobile devices”, *Media Watermarking, Security, and Forensics 2013*, Vol. 8665, p. 866507, 2013.
- [54] J. M. Jobe and M. J. J. o. t. A. S. A. Pokojovy, “A cluster-based outlier detection scheme for multivariate data”, *Journal of the American Statistical Association*, Vol. 110, No. 512, pp. 1543–1551, 2015.