Supervised Models for Loan Fraud Analysis using Big Data Approach

Girija Attigeri, *Member, IAENG,* Manohara Pai M M, Radhika M Pai,

***Abstract*—Banking and Financial Institutions are facing the pressure of increased defaults by individuals and firms in the last few years repercussions due to fraudulent activities. It is not only adversely affecting banks but also other financial sectors which depend on them. This makes it imperative to study the ways to prevent them rather than curing the situations. However, banks face two challenges in identifying NPAs and Wilful defaults. The first one is the due diligence of firms/individuals before an extension of the loan. The second one is, need for the placement of automated safeguards to reduce frauds originating out from human behavior. The wilful defaults are committed mainly in loan and credit services for personal benefits and are getting converted into bad loans. Bad loans are the Non-Performing Assets (NPAs) and wilful defaults are a subset of these. Hence, it is very important to control NPAs. The objective of the paper is to design and evaluate machine learning based supervised models for NPA detection. To design models, the entire historical and current data needs to be considered, which requires, faster access to large volumes of heterogeneous data. Hence, the supervised models are implemented using big data techniques for fraud detection and analytics. The various supervised models namely Logistic Regression, Support Vector Machine, Random Forest, Neural Network, and Naive Bayes are designed for loan data and experimented using Map Reduce on Hadoop platform. These models are evaluated considering various performance metrics. The empirical result shows that the Neural Network model performs best considering precision, recall, relative commission error, and kappa statistics for NPA prediction. The best-performed model can be integrated into the existing loan management system for the early identification of NPA cases.**

***Index Terms*—Loan Frauds, Non-Performing Assets, Machine Learning, Supervised Models, Big Data Approach, Hadoop Platform.**

I. Introduction

ACCORDING to The Wire [1], Indian banks reported around 8670 loan fraud cases amounting to Rs. 612 billion from 2013 to 2018. The PWC, Delloite surveys [2], [3], and RBI reports [4] indicate frauds in banks as one of the emerging risks for the financial sector affecting the economy of the country. According to the Economic Times [5], 18 public sector banks have faced a loss of Rs. 32000 Crores. in the first quartile of 2019 and most of these are due to bad loans. Bad loans are the Non-Performing Assets (NPAs) which are used as one of the measures for assessing the financial health of a bank. The Last 10 years have seen huge numbers of NPAs due to which the government has focused on changing the policies for controlling the same and has brought Insolvency and Bankruptcy Code [4]. However, banks do not have a strong governing and monitoring process to analyze and detect NPAs at the earliest. Banks need an automated process to capture and analyze the data for early detection of NPAs to avoid losses.

Early detection of NPA needs complete data analytics by considering the entire historical and current data. This includes data about both Performing Assets (PA) and NPA cases. To distinguish NPA and PA cases, understanding the attributes and common patterns related to these cases are necessary. The relation between the attribute values, patterns, and the NPA cases must be identified. This can be done by building automated models which can establish these relationships to identify loan frauds (NPA) for new cases in real-time. These automated NPA detection techniques can help to detect new NPAs as well as prevent them at the earliest. A financial organization that has an efficient loan fraud detection system can save money and protect businesses, their employees, shareholders, and customers. This can also help in identifying loan frauds occurring due to insider involvement, thereby reducing costs associated with the fraud, improving financial, operational results, and maintaining stakeholder's confidence.

There are various techniques like Data Mining, Expert Systems, Rule-based Approach, Model-based Reasoning, and others, which are currently applied to detect fraud. These techniques are constrained by conventional data processing capabilities. Conventional approach for fraud detection is based on structured data processing, but new generation fraud cases deal with unstructured data. Hence, fraud detection model needs to deal with capturing and processing of unstructured data, keeping it conversant with the new fraud incidences. Therefore, a big data approach is needed. Hence, the paper focuses on building efficient Machine Learning (ML) based classification model for predicting fraudulent activities in loan stream data using big data technologies, making it suitable for an adaptive fraud detection system.

The ML-based models can be unsupervised, supervised, or semi-supervised. If there is sufficient knowledge available for the NPA cases then supervised models can be used. Unsupervised models are used when sufficient knowledge is not available regarding PA and NPA cases. Therefore data is grouped based on similarity and these groups are labeled as PA and NPA based on experts advice. The current work focuses on the use of supervised models as the data labeled as PA and NPA by the experts is available. Classification algorithms are used to build supervised models to detect NPAs. Training data consists of samples and their corresponding class labels. The class labels in the proposed fraud detection

Manuscript received February 12, 2021; revised August 24, 2021.

Girija Attigeri, Assistant Professor, Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education. Manipal, India, 576104, e-mail: (girija.attigeriv@gmail.com).

Manohara Pai M M, Professor, Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education. Manipal, India, 576104, e-mail: (mmm.pai@manipal.com)

Radhika M Pai, Professor, Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education. Manipal, India, 576104, e-mail: (radhika.pai@manipal.com)

system, indicate whether an observation (loan account) is NPA or PA. Once the accounts are classified PA or NPA, further analysis can be carried out to understand whether the NPA cases are wilful or not.

The supervised models have two phases; training and testing. The labeled samples are used to build the prediction model and the built model is validated in the validation phase. The validated model can be used to predict the class labels for the new samples. There are various ML algorithms available for classification. The challenge is to select the best suitable model that can detect NPA efficiently. To build such a model, faster access to large volumes of heterogeneous data is needed. But the available research does not focus on such big data analytics. The proposed research aims at big data insight for fraud analytics and detection.

The rest of the paper is organized as follows. Section 2 presents background NPA, Machine learning, and other techniques for fraud detection algorithms. Section 3 describes the methodology, Section 4 explains the implementation of NPA detection using the Big Data Approach on spark framework, and empirical results are discussed in Section 5. Finally, Section 6 summarizes and concludes of the paper.

II. Background

Literature related to fraud detection in the banking domain, NPA, and wilful defaults is surveyed for technologies, processes, and limitations. Many researchers [6],[7],[8],[9], and [10] worked towards loan fraud identification and have analyzed lender-borrower relationships. The authors in [6] used knowledge graphs and machine learning algorithms to identify the potential candidates with an intention to commit fraud. Wang et al.[8] used Random Forest and gradient boosting decision tree along with Principal Component Analysis (PCA) to detect frauds in the financial market. The financial health of the customers of a bank was analyzed using the Hidden Markov Model (HMM) and feature-based clustering by Philip et al. [9]. The authors in [11] and [12] conducted a study on wilful defaults in the Indian scenario. Satish et al. in [11] presented a study to relate wilful defaults and moral hazards.

Authors in [13], [14], [15], [16], [17], [18] and [19] investigated banking sectors of Europe, Asia, US, and Africa to find the reasons for NPA. The authors listed bank-specific factors as growth in loans, return on equity, return on asset, various financial ratios, real interest rate, inflation rate, and economic growth. The data considered was the gross NPA. The relation of identified factors with gross NPAs was established using panel regression models.

The authors in [20], [21], [22] discussed various methods on Outliers Detection for Financial Transactions in their survey. Some of them are listed below.

Clustering-based approach: The data points are grouped based on similarity. Points that do not belong to any cluster or not forming small clusters are considered outliers.

Model-based approach: A model is used to represent normal data. Outliers are points that do not fit in the model [23].

Rule-based approach: It defines a set of rules to describe normal data points. The points which deviate from the rules are considered outliers.

A conceptual real-time fraud detection model in the banking sector was discussed by John et al. [24]. The authors discussed data mining techniques such as association, clustering, classification approaches for fraud detection. The authors mined synthesized data for association rules based on marital status and range of withdrawal amounts. The authors emphasized continuous analysis of various sources of data and the use of suitable technology.

Sanchez et al. [25] proposed a conceptual framework using the fraud triangle theory for identifying and outlining a group of people inside an organization, who commit fraud. The authors emphasized the use of advanced technologies for capturing data, and for analyzing user behavior continuously.

The authors in [26] used supervised NPA detection algorithms. Local Outlier Factor (LOF) and Isolation forest were implemented for ATM fraud detection. The authors used 100000 transactions out of which 227 were identified as fraud. The authors showed that isolation forest performed better than LOF.

The authors in [27], [28], and [29] provided a survey of supervised and unsupervised NPA detection algorithms. Malini et al. [27] discussed the use of Logistic Regression, Decision tree, Neural network, Hidden Markov Model, Support vector machine, and k-Nearest Neighbors (kNN) for credit card fraud detection. The authors provided comparisons on the parameters handling the non-linearity of the data and real-time detection of the fraud. The authors claimed that kNN based NPA detection methods suited well for credit card fraud detection with limited usage of the memory. Carcillo et.al [29] concluded that the combined techniques worked better than the individual approaches.

Rule-based approaches and Machine Learning based classification Techniques were used for identifying fraud involving transaction data in [30] by Wee-Yong et al. In the first technique, predefined rules were applied to qualify these NPAs as fraudulent or not. The second technique involved the application of classification techniques such as Neural networks, SVM, etc.

Credit card expenditure data was simulated using Spark for analyzing fraudulent behavior by Sathyapriya and Thiagarasu[31]. The data samples were clustered into three risk groups namely low, medium, and high. Further, HMM was used to generate the observation symbol and was compared with the threshold value. If it was less than the threshold, the transaction was considered genuine, otherwise fraudulent.

Xu et al. in [32] discussed about loan fraud detection in P2P set-up. The authors listed the research questions on the dataset, features to be considered, techniques, and applicability of Big data approach. The study concluded that for fraud detection, user behavior and transaction histories need to be analyzed and Big data approach can be used for this effectively and efficiently.

Authors in [33] and [34] discussed machine learning algorithms for clustering namely traditional k-Means, Fuzzy k-Means, and Streaming k-Means and classification algorithms namely Logistic Regression, Naïve Bayes, Random Forest, Hidden Markov Models, and Multilayer Perceptron using the Hadoop ecosystem and Spark. Ferhat et al., discussed MapReduce-based distributed SVM algorithm for binary classification.

In summary, the papers indicate the importance of considering several factors such as payment behavior, social behavior, financial, demographic, professional data for analysis of NPA and wilful defaults. The papers emphasize on use of Big data techniques and tools for fraud detection considering heterogeneous data from various sources. However, the papers show that there is not much effort in using Big data analytics for NPA/Wilful default detection. The papers indicate that various data mining, machine learning, and outlier detection techniques used for credit card fraud detection, insurance fraud, intrusion detection, and others can be used for loan fraud detection and prevention as well. However, the literature shows that the research effort towards loan fraud detection needs to be put to detect NPA/Wilful default at the earliest to save the financial institutions from facing damage.

The goal of the present research work is to design and evaluate supervised machine learning models for loan fraud detection. Loan Frauds are identified by using advanced technologies mainly focusing on NPA/wilful default behavior in the banking sector. Assuming the relevant data is collected and data preparation is done for analysis, some of the research questions for NPA detection are:

1) Which methods/techniques can be used to efficiently detect NPA/Wilful default?: Use of advanced techniques for the identification of loan fraud at the earliest can be employed effectively.
2) How can Big data technology be leveraged to detect NPA/Wilful default?: The broad perspective of NPA/Wilful default analysis considers all customers of the bank from various sources possible. This data is voluminous and heterogeneous, hence Big data technology needs to be used.
3) How to build an integrated system for detecting NPA/Wilful default?: Various phases of the fraud detection, beginning from data generation, preprocessing, fraud detection, and visualization need to be put in an integrated system.

III. Proposed Work

The data to be considered for loan fraud detection comes from heterogeneous sources such as weblogs, social media, organization internal database, etc. Data could be structured or unstructured. These data need to be combined for analyzing fraud. The data to be analyzed is stored in a Hadoop store. Historical data of the banking system is required to be analyzed to get the patterns of normal behavior. The raw data available is not ready for analysis. It is preprocessed by cleaning, removing the noise, normalizing, and reducing dimension. While preprocessing, it is important to identify parameters that help in identifying fraud. These parameters are used for loan fraud detection using machine learning algorithms. Loan fraud detection involves two main components: 1) unsupervised NPA detection component 2) supervised classification. The first NPA detection component is based on unsupervised approach. It begins with all unlabeled data objects. Based on the domain and loan parameters identified, the measures suitable for detecting outliers have to be designed. A supervised classification model is built when the labeled data is available. Here a predictive model is trained for classifying an activity as fraud or not. The main focus of this research is to build and evaluate various classification models using Big data technology for identifying suspicious activities (outliers) in the financial domain and generate a trigger to avoid losses due to fraudulent actions.

The process flow for loan fraud detection is shown in Figure 1. The process is intended to analyze the entire data available to detect outliers that are fraudulent. It takes entire data and adaptive knowledge base as the input in the first phase for detecting outliers (loan frauds) using unsupervised approach. In the second phase, the new data object is assigned an outlier score by checking its values for the fraud parameters identified. If the outlier score is higher than the threshold value, it is considered a potential fraud (outlier). These cases are sent for further confirmation to the supervised classification component which uses an adaptive knowledge base, which further denies or confirms them as fraud. For supervised classifiers, methods and measures suitable to the domain have to be designed. The classifier needs to be trained with the entire data available. It is intended to make use of an adaptive knowledge base, fraud repository, and normal database for analyzing whether the given input is fraud or not. The main focus of this research is to evaluate the supervised models for fraud detection.

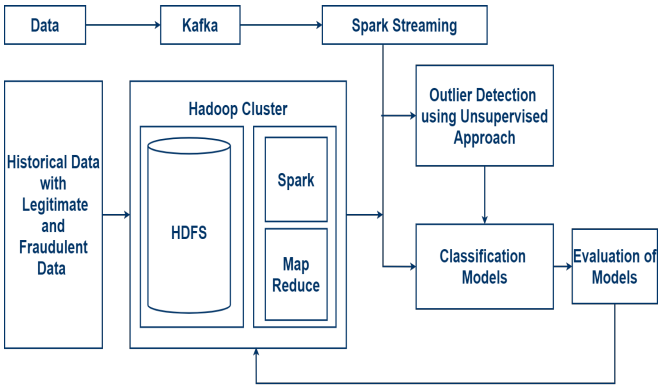

Fig. 1. Process Flow of NPA Detection using Supervised Model

The technology stack used in the research is shown in Figure 2. At the basic level data store and processing is carried out on the Hadoop platform. In the second level Apache Spark and Kafka are used for simulating real-time stream computing. The data analytical components data collection, data preparation, building machine learning models, and visualization are on the top layer. The Big data approach is implemented by writing map-reduce logics for supervised models on Hadoop Cluster considering three nodes.

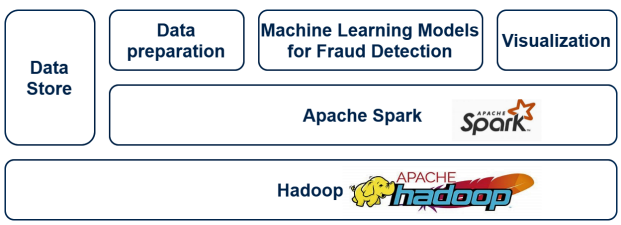

Fig. 2. Technology stack used for Fraud Detection

IV. MapReduce Algorithms of Supervised Models for NPA Detection

In this section, the design of parallel distributed classification algorithms is presented. The implementations of these are carried out using Hadoop. For each of the classification algorithms, parameters are varied to get the efficient model such as the structure of the neural network, kernel function for SVM, hypothesis functions for regression, and decision criteria for the random forest. These models are designed using Map and Reduce functions. The models are compared based on the accuracy obtained, and the algorithm with the best accuracy is considered for prediction in the fraud detection system.

A. MapReduce-Naive Bayes

Naive Bayes model is a generative model built for prediction using prior and likelihood probabilities as shown in equation 1.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

where

$P(B|A)$: Likelihood probability: Probability of the given input variable B given class label A.

$P(A)$: Prior probability: Probability of a given class label A.

$P(B)$: Predictor prior probability : Probability of a given input variable B.

Outcome variable is *loan_status* and selected 60 features are independent variables. For example, considering the independent variable *purpose* with value *'carloan'*, the chances of *loan_status* being 'PA' or 'NPA' are computed as:

loan_status = P(*loan_status* = 'PA') * P(*purpose* = 'carloan' | *loan_status* = 'PA') / P(*purpose* = 'carloan')

loan_status = P(*loan_status* = 'NPA') * P(*purpose* = 'carloan' | *loan status* = 'NPA') / P(*purpose* = 'carloan')

where P(*purpose*='carloan'|*loan_status*='PA') is likelihood probability.

P(*loan_status*='PA') is class prior probability.

P(*purpose*='carloan') is predictor prior probability.

The status is predicted as the one with a higher probability.

The MapReduce-Naive Bayes (MR-NB) model is designed to estimate these prior and likelihood probabilities of a given class label [35]. The Mapper takes the row as input and converts it into key-value pair, where, *key* consists of a class label, attribute name, and attribute value. The *value* in key-value pair is set to 1 for an individual occurrence of the key. For example if *label*='PA', the attribute name is *purpose* and attribute value is 'business' then the key will be < key='PA'+*purpose*+'business'> and value will be $< 1 >$. Hence, the key-value pair is <'PA'+*purpose*+'business', 1>. The Reducer aggregates each unique key value to compute the probabilities. The examples of unique key are {'PA'+ *purpose*+'business'}, {'PA'+ *purpose*+'carloan'}, etc. Proportions of each unique value of the feature with the corresponding label is the likelihood probability *Pr*. The outcome of the Reducer is likelihood probability *Pr*. Prior probabilities are proportions of 'NPA' and 'PA' records in the dataset. Driver function uses these probabilities to predict the labels for the test samples. The predicted and actual labels are used to compute the confusion matrix for Naive Bayes. The steps of Naive Bayes are shown in Algorithm 1.

Algorithm 1 MapReduce Naive Bayes

```
1: procedure MAPPER(Data with all Features, Class Label)
        ▷ Data: Loan data with 60 features, Class Label: PA/NPA
2:     Read Dataset
3:     for each Loan record do
4:         for each Attribute do
5:             Emit(Loan_Status+Attribute_Name +Attribute_Value, 1)
6:         end for
7:     end for
8: end procedure
9: procedure REDUCER(Loan_Status +Attribute_Name+Attribute_Value, Iterator)
10:    for each value in the Iterator do
11:        Pr = Pr+ value
       ▷ Pr: Probability of Attribute value and class label
12:        Emit(Loan_Status+Attribute_Name +Attribute_Value, Pr)
13:    end for
14: end procedure
```

B. MapReduce-Logistic Regression

The hypothesis function in Logistic Regression relates exogenous variables which are 60 selected features and the outcome variable which is the status of the loan. The model returns an estimated probability score of predicting the status as 'PA' or 'NPA'. This probability score is given as input to the Sigmoid activation function. If the input probability value is zero or positive, then the prediction will be a value greater than or equal to 0.5, which is approximated to 1 as 'NPA'. A negative input probability value returns a value less than 0.5, which is approximated to 0 as 'PA'.

The MapReduce-Logistic Regression (MR-LR) algorithm is designed for fitting a linear equation [36]. Mapper and Reducer functions for carrying out the computations are shown in Algorithm 2.

The Mapper and Reducers are designed to fit the Logistic Regression model using gradient descent. It gives *theta* values that represent the model parameters of Logistic regression. The Mapper updates intermediate *theta* values $temp\_\theta$, considering the dataset in each DataNode. The Mapper function emits attribute number and corresponding intermediate theta value as <j, $temp\_\theta_j$ >. For example, considering *loan amount* and its *theta* value, key-value pair will be $< loanAmount,\ temp\_\theta_{loanAmount} >$. These intermediate *theta* values are combined in the Reducer to compute the final *theta* values for each attribute. MapReduce is iteratively executed in the driver function until the convergence criterion is met or a maximum of 50 iterations. It uses the sigmoid activation function to convert real values between 0 to 1, which can be used for binary classification. The threshold is set as 0.5 and the values greater than the threshold are considered as NPA. Finally, the confusion matrix is obtained in the Driver function.

C. MapReduce-Support Vector Machine

Support vectors are data points closer to the plane and influence the orientation along with the position of the

Algorithm 2 MapReduce Logistic Regression

procedure MAPPER(Data with all Features, Class Label, Initial θ values)
▷ Data: Loan data with 60 features, Class Label: PA/NPA
  Read Dataset
  for each Attribute j **do**
    for each Loan record i **do**
      Compute $temp\_\theta_j^{(i)} = \sum (h_\theta (x^i) - y^{(i)}) x_j^{(i)}$
    ▷ $temp\_\theta_j^{(i)}$: intermediate θ value updation for j^{th} attribute considering i^{th} sample,
    ▷ h_θ: hypothesis function to predict label using intermediate θ values,
    ▷ $x_j^{(i)}$: j^{th} Attribute value in i^{th} sample
    end for
    Emit(j, $temp\_\theta_j$)
  end for
end procedure
procedure REDUCER(Key Attribute j, Iterator of temp values for each attribute j)
  for each attribute j **do** compute
    $\theta_j = \theta_j - \frac{1}{count} \sum temp\_\theta_j$
    Emit(j , θ_j)
  end for
end procedure

Algorithm 3 MapReduce Support Vector Machine

procedure MAPPER(Data with all Features, Class Label, Initial Weights W)
▷ Data: Loan data with 60 features, Class Label: PA/NPA
  Read Loan Dataset
  for each Loan record x_i **do**
    $f_i = W^T * x_i + b$
  ▷ f_i is decision value $f_i > 1$ or $f_i <$ -1 for the valid support vector x_i
    $H1$ = f_i+1
    $H2$ = f_i-1
    SV = Data Point present on H1 and H2
    ▷ SV:Local support vectors for data on the node and $H1$ and $H2$ are supporting boundary lines of hyperplane
    $W = W$ + α * sigmoid(f_i) * x_i ▷ Update W
    $y_i = Sigmoid(f_i)$ ▷ To predict the label
  end for
  Emit(1, W, SV, b)
end procedure
procedure REDUCER(key 1, Iterator of W, SV,b))
  Global b = $\sum_{i=0}^{count} b/count$
  Global W = $\sum_{i=0}^{count} W/count$
  Global SV += SV
  Emit(Global W, Global SV, Global b)
end procedure

hyperplane. Hyperplane for the NPA prediction is built such that it separates the samples, which are labeled as 'PA' and 'NPA'.

The MapReduce-Support Vector Machine (MR-SVM) algorithm is designed to compute support vectors using linear kernel function, which clearly transforms the feature space and segregates the two classes of the training sample [37]. A hyperplane is represented by weights *W* and bias *b*, and Support Vectors *SV* and are the model parameters. In the Mapper function, the local support vectors are calculated for the data on each node. Initial weights and bias are randomly assigned. For these weights and bias values, initial hyperplane and boundary planes are computed. Local support vectors are the actual data points present on boundary planes. Based on the predictions considering the hyperplane, weights are updated. Then using the model parameters the predictions are computed for data residing in the node. The predictions are '0' for 'PA' and '1' for 'NPA'. Here the key is considered as 1 and the value consists of Weight *W*, local support vectors *SV* and Bias *b*. Mapper function emits <1, value>. The Reducer function merges these values to compute global *W*, *b*, and *SV* and emits the same.

The Driver function calls Mapper and Reducer functions iteratively until the convergence criterion is met. The convergence criterion is to minimize the loss function. The steps of MR-SVM are shown in Algorithm 3. Finally, the computation of the confusion matrix is carried out in the Driver function.

D. MapReduce-Neural Network

Neural Network (NN) is a model of artificial neurons arranged in multiple layers [36],[38]. If the model learns through the concept of back-propagation then it is called as a Back-propagation Neural Network (BNN). The data is iteratively fed to the input nodes, computations are carried out through hidden layers and the result is obtained in the output layer node. This output is compared with the actual labels to compute the potential error. This computed error is fed back to the model in each iteration. Based on the diminishing error and desired output, the weights are adjusted and the final model is obtained.

Predicting NPA is the classification task performed on labeled data. Hence, it does not require more layers in the NN. To decide the number of hidden layers and neurons in each hidden layer, the literature suggests splitting the data into training and test data to train the NN starting from 1, 2, and 3 hidden layers. Whichever gives the low computation cost can be chosen as a final NN structure. Also, the number of neurons in each hidden layer can be computed by using Equation 2.

$$N = ((InputNodes + OutputNodes) * 2)/3. \quad (2)$$

where $N \rightarrow$Number of nodes

To determine the number of hidden layers, the experimentation has been done and identified that one hidden layer with 40 nodes as per Equation 2, is the NN structure to train the model. However, the accuracy obtained was low for one hidden layer network as compared to two hidden layers. The computation of neurons for two hidden layers is

performed and the final structure of NN is identified as shown in Figure 3. The NN structure contains an input layer with n=60 neurons for 60 identified features, hidden layer 1 with m=40 neurons, hidden layer 2 with p=27 neurons, and an output layer with 1 neuron to identify the 'PA' or 'NPA'.

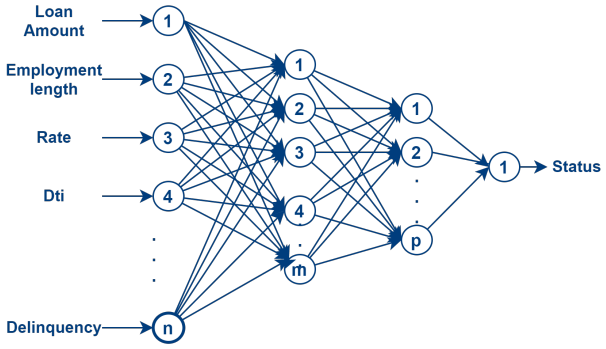

Fig. 3. Neural Network Structure

Algorithm 4 MapReduce Neural Network

procedure MAPPER(Data with all Features, Class Label)
▷ Data: Loan data with 60 features, Class Label: PA/NPA N: Number of Records, d: Dimension, h number of hidden layer, W: Randomly assigned Initial Weights
  Read Dataset
  for each hidden layer h **do**
    for each attribute a_i and neuron j in the hidden layer **do**
      Compute $I_j = \sum_{i=1}^{d} a_i * w_{ij}$
    ▷ Intermediate value computed in j^{th} neuron in h
      Compute $O_j = Sigmoid(I_j)$
  ▷ Outcome of j^{th} neuron in h after sigmoid activation
    end for
    for Each neuron j in the hidden layer h **do**
      Compute $I_o = \sum_{j=1}^{h} O_j * w_{jo}$
      Compute $O = Sigmoid(I_o)$
      $Err_O = O(1-O)(T-O)$ ▷ Compute Error term at Output layer
    end for
    for Each neuron j in the Hidden layer **do**
      $Err_j = O_j(1-O_j)Err_o$ ▷ Compute Error at neuron j in the Hidden layer
      $W_{ij} = W_{ij} + Err_j * O$
    end for
  end for
  Emit(1,$\{T, O\}$)▷ T: Actual Class label, O: Predicted Class label
end procedure
procedure REDUCER(Iterator of T and O)
  Compute evaluation metrics;
  ▷ Computing True Positives, True Negatives, False Positives, False Negatives, Accuracy
  Emit (Metric, Computed Value of the metric)
end procedure

The MapReduce-Neural Network (MR-NN) algorithm is designed to use a Back-propagation Neural Network (BNN) at each Mapper and combines the outputs at the Reducer stage. First, input to the Mapper function is a Loan data file with 60 features and labels. This file is read by the Mapper as key-value pair where the key is offset to the file and the value is each loan record from the file. Second, input is the initial weights for the NN. The Mapper computes intermediate values at each neuron in the hidden layers. These intermediate values are computed by applying the sigmoid function on the weighted sum of all the inputs given to the neuron. These intermediate values provided by the last hidden layer are used to compute the predictions at the output layer. Using these predictions and actual outcomes the error at the output layer is computed. This error is back propagated to adjust weights to diminish error. With the updated weights, classification is performed on each of the data blocks. The Mapper emits $<1, \{Actual\_label, Predicted\_label\}>$. In the Mapper, key is set as '1' to consider all the values together for the Reducer function. The Reducer combines all the key-value pairs and computes the confusion matrix having true positives, true negatives, false positives, and false negatives. Based on confusion matrix Reducer computes accuracy and emits output as <Metric, Computed_Value>, for example $< accuracy, 94.7 >$. The driver runs the Mapper and Reducer functions in the loop until convergence. The steps of Mapper and Reducer functions are shown in Algorithm 4.

E. MapReduce-Random Forest

Random Forest is a collection of decision trees. Each decision tree predicts the outcome as 'PA' or 'NPA'. These decisions are combined using majority voting by counting the decisions.

The MapReduce-Random Forest (MR-RF) is designed for NPA prediction. The Mapper function works on the dataset at the node to get the decision tree, i.e., identify decision attributes at each level by computing the entropy, and information gain. Initially, the entropy of the whole sample is calculated. Subsequently, entropy is computed for each value of the attribute, using these entropies final entropy of the attribute is computed. Using these computations Information gain for the attribute is calculated [39].

The decision node is the attribute with maximum information gain. The data is split based on the decision node attribute. The Mapper function emits the constructed Decision Tree (DT). The key is set as '1' and the value is constructed *DT* at that node, hence, key-value pair emitted is $< 1, DT >$. For example, decision tree *DT1* is the tree with root as *loan_amnt*, second level attribute as the *purpose* and so on. The Reducer function uses these decision trees for prediction on the test data. The decisions of all these trees are combined using majority voting, and the results are emitted from the Reducer. The key-value pair emitted from Reducer is $< sample\ i, Prediction >$. The driver function uses this output to compute the confusion matrix. The steps of Mapper and Reducer functions for Random Forest are shown in Algorithm 5.

V. EXPERIMENTATION

The data considered for experimentation is loan data set having approximately more than two lakh instances with

Sr. No.	Name of Parameters	Sr. No.	Name of Parameters	Sr. No.	Name of Parameters
1	acc_now_delinq	25	inq_last_6mths	49	out_prncp_inv
2	addr_state	26	installment	50	policy_code
3	all_util	27	int_rate	51	pub_rec
4	annual_inc	28	issue_d	52	purpose
5	annual_inc_joint	29	last_credit_pull_d	53	pymnt_plan
6	application_type	30	last_pymnt_amnt	54	recoveries
7	collection_recovery_fee	31	last_pymnt_d	55	revol_bal
8	collections_12_mths_ex_med	32	loan_amnt	56	revol_util
9	delinq_2yrs	33	loan_status	57	sub_grade
10	desc	34	max_bal_bc	58	term
11	dti	35	member_id	59	title
12	dti_joint	36	mths_since_last_delinq	60	tot_coll_amt
13	earliest_cr_line	37	mths_since_last_major_derog	61	tot_cur_bal
14	emp_length	38	mths_since_last_record	62	total_acc
15	emp_title	39	mths_since_rcnt_il	63	total_bal_il
16	funded_amnt	40	next_pymnt_d	64	total_cu_tl
17	funded_amnt_inv	41	open_acc	65	total_pymnt
18	grade	42	open_acc_6m	66	total_pymnt_inv
19	home_ownership	43	open_il_12m	67	total_rec_int
20	id	44	open_il_24m	68	total_rec_late_fee
21	il_util	45	open_il_6m	69	total_rec_prncp
22	initial_list_status	46	open_rv_12m	70	total_rev_hi_lim
23	inq_fi	47	open_rv_24m	71	url
24	inq_last_12m	48	out_prncp	72	verification_status

Fig. 4. Features of the Loan Data

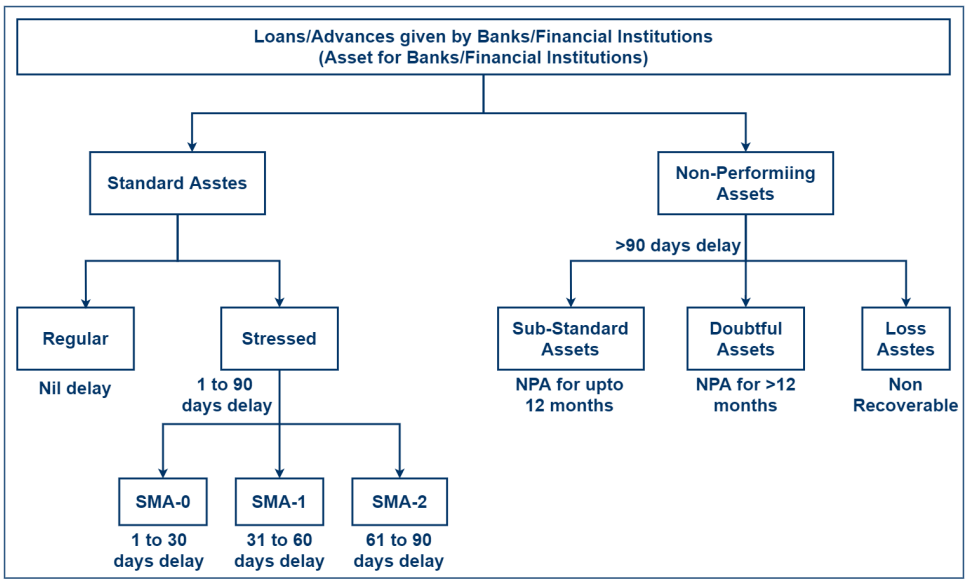

Fig. 5. Asset Classification (Source: https://www.rbi.org.in/scripts)

sixty selected features shown in Figure 4. The sample features are *loan_amnt, installment, emp_length, credit_score, , purpose, designation, delinquency, house type,* and others. The sample data is shown in Figure 4. The dataset has around 5% of instances are NPA and the remaining are PA. The dataset is divided into a training set of 140000 instances, a test set of 55996 instances, and a validation set of 5004 instances. In the dataset, the loan_status is the class label which has values "current" if payments are not delayed, "fully paid" if the loan is completely paid, and "charged off" if there payment delays. As per RBI's asset classification, as shown in Figure 5, an asset is classified as Special Mention Accounts (SMA) if there is a delay of less than 90 days in the loan payments. If the delay is more than 90 days it is put under the NPA category. Taking this into account, the status of the loan is factored as PA and NPA. If the values of status are either current or fully paid, it is factored as PA and if it is late or charged-off, then it is NPA.

For building NPA detection models big data platform Hadoop is used. It enables the distributed processing of large data sets across clusters. These clusters are formed using commodity servers. It is a platform that provides distributed storage and computational means. Hadoop works on the principle of distribute-join-aggregate. In a fraud detection system, considering the loan analysis scenario, the data samples with relevant features are distributed across multiple nodes in the Hadoop cluster. For example, to compute the monthly average spending amount of customer A, there is a need of analyzing all the debit transactions of A for a month. This analysis shows the average spending pattern of

Fig. 6. Spark Kafka Integration

Algorithm 5 MapReduce Random Forest

procedure MAPPER(Data with all Features, Class Label)
▷ Data:Loan data with 60 features, Class Label: PA/NPA
  for each Record in Dataset D **do**
    for each attribute A_i **do**
      $E(D) = -\sum P(A_i)\log_2 P(A_i)$ ▷ Entropy of attribute A_i
      $IG(D, A_i) = E(D) - \sum everyvaluev \in A_i \frac{Numberofvaluesfor A_i}{Numberofsamplesin D} E(A_i = v)$
      ▷ Compute Information Gain (IG_i) for A_i
      A = A_i having max(IG_i) ▷ A is a decision attribute having maximum IG
    end for
    $DT \leftarrow (A)$ ▷ DT:Decision tree
    Split data based on values of attribute A
  end for
  Emit(1, DT)
end procedure
procedure REDUCER(1, Iterator DT)
  D: Read Test set
  for each Record i in D **do**
    for each DT_j in the Iterator **do**
      $Result_{ij} \leftarrow$ Predict the class using DT_j
      if $Result_{ij}$ = 'PA' **then** ▷ 'PA' and 'NPA' are class labels in Loan data
        $Count(PA) += 1$
      else
        $Count(NPA) += 1$
      end if
    end for
    Prediction= max($Count(PA), Count(NPA)$)
    Emit(Record i, Prediction)
  end for
end procedure

the customer. This data is split by the NameNode across the DataNodes in the HDFS cluster. Further, data has to be joined for getting complete data of customer A's cluster. The processing logic is written as key-value pairs in MapReduce functions. Here, customer A is the key, and $AmountSpent$ for each transaction is the value for each map function in the TaskTracker node, such as key-value for customer A $\{(A, 1000), (A, 456), (A, 14562), (A, 9876), ...\}$. Subsequently, the intermediate sort and shuffle phase of the Hadoop combines all the values having same key, such as $\{(A, \{1000, 456, 14562, 9876, ...\}), \}$. This goes as input to the reducer. The Reducer aggregates the values and computes the average for the same key. Further, it emits $< A,\ AverageAmountSpent >$. The output of the reducer is stored in the HDFS file system for further analysis. The $AverageAmountSpent$ is compared with Annual Income, if it is more, then this feature is contributing towards identification of NPA. Similar logic is written for other MapReduce functions to analyze features of the dataset such as *number of installments, investments, purpose, delinquencies, write-offs*, etc. The final result of all the MapReduce functions is combined for each customer and is further given for the detection of outliers such as NPA.

Supervised algorithms are implemented on the Hadoop platform using Spark and Kafka for simulating a real-time environment for stream data. Kafka is used for building data pipelines for streaming applications. It is a message broker system used for providing streaming data to Spark as shown in Figure 6.

The Spark engine is used for implementing the parallel distributed classification algorithms. The loan data stored in the HDFS is read by the Kafka message broker system and fed to the Spark streaming application as a stream of loan instances. It is further converted to a batch based on the windowing concept and given to the classification algorithms for training and testing. The models are validated based on the evaluation metrics and if improvements are required, then the process of training the model is repeated. These classification models are compared and the best one is used in the final framework for NPA detection.

VI. Results and Discussions

Exploratory data analysis is performed on the Distribution of NPAs Recovered by Banks presented in the Report on Trend and Progress of Banking in India [40]. Figure 7 shows the total NPA amount from 2003-04 to 2019-20. It can be observed from the graph that the amount involved in NPA is increasing exponentially and the amount recovered is significantly low. Figure 8 shows the total amount lost due to fraud cases in banks. It can be observed that fraud related to advances is contributing most to the frauds. Sector-wise gross NPA analysis is shown in Figure 9. It can be observed from the graph that Scheduled Commercial Banks (SCB) and public sector banks are affected by NPAs the most. In that the loans offered to private sector, agriculture, micro and macro finances have turned NPA. This analysis gives an insight into the significance of detecting NPAs and wilful defaults at the earliest. In order to perform the same, experimentation is performed on around 2 lakh instances having 5% of instances labeled as 'NPA' and the remaining as 'PA'. The training, test and validation sets consists of 140000, 55996 and 5004 loan records respectively. The Confusion matrices obtained with validation data for Logistic Regression, Neural Network, Naive Bayes, Random Forest, and SVM are shown in Figure 10. Confusion matrix for binary classification model consists 4 values, True positive: The sample predicted as NPA and its actually NPA, True Negative: The sample predicted as

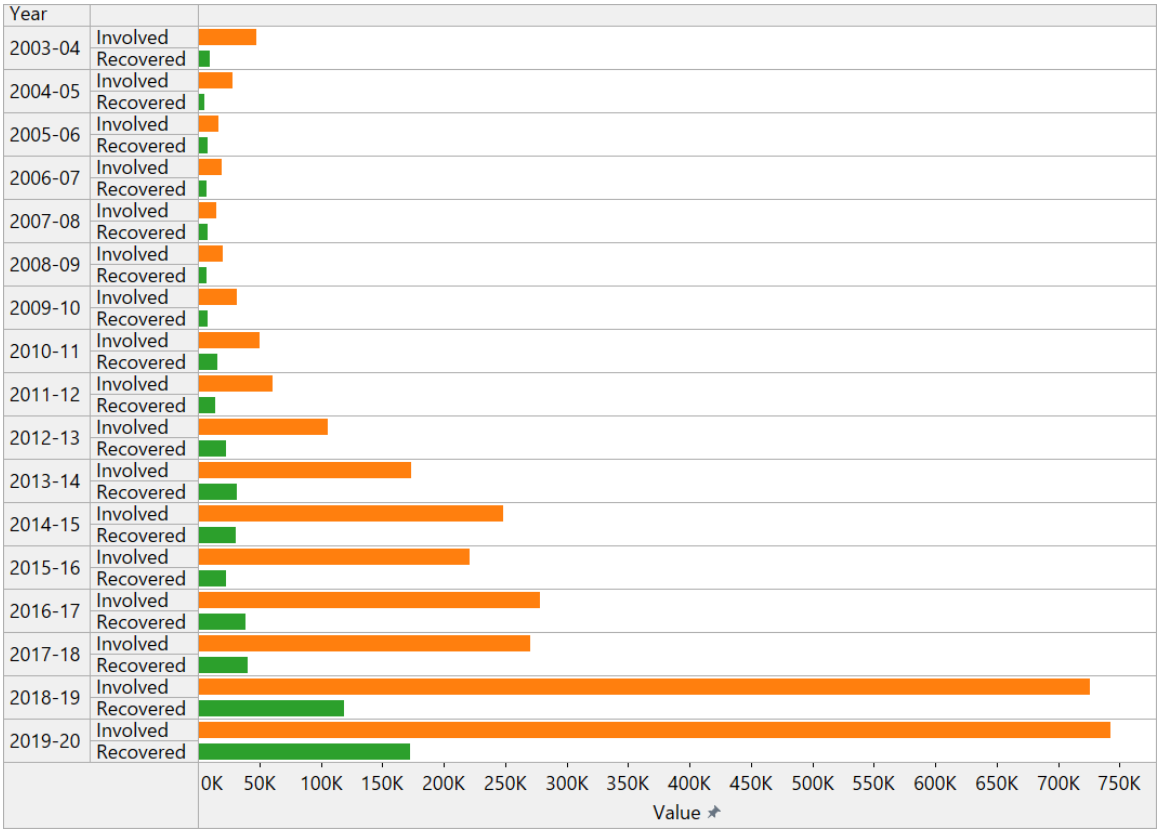

Fig. 7. Distribution of NPAs Recovered by Banks

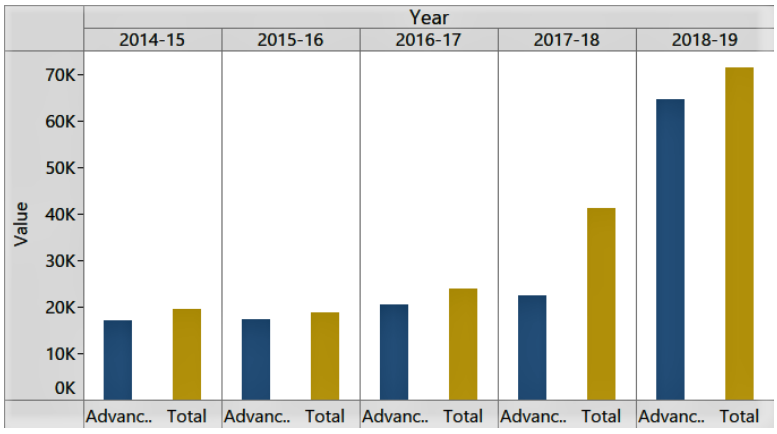

Fig. 8. Total amount of fraud and fraud related to advances

PA and its actually PA, False Positive: The sample predicted as NPA and but its actually PA, False Negative: The sample predicted as PA and but its actually NPA. These values of confusion matrices are used to compute various evaluation metrics such as accuracy, precision, recall, f1-score, and kappa statistics. Figure 11 shows the accuracies of each of the models. Accuracy is computed by dividing the number correctly classified samples by total number of samples. The results in the Figure 11 depict that MR-Neural Network has the highest accuracy. Accuracy depicts how well the model performs considering true positives and true negatives. However the model needs to be evaluated with respect to false positive and false negatives (misclassification) as well, Hence other metrics have been used to evaluate a model completely for NPA prediction problem.

Evaluation of classification models using Precision, Recall, F1-score, Specificity, and Kappa statistics is shown in Figure 12. Precision represents the measure of correctness of predictions concerning predicted positive examples i.e the examples predicted as NPA. Higher Precision indicates low False positives i.e. small number of examples are incorrectly classified as NPA. Neural Network has the best precision compared to other models, indicating very few PA cases are classified as NPA. Recall or Sensitivity gives the measure of correctness of predictions concerning actual positive examples i.e actual NPAs. The results show that all the models have almost the same recall rate. F1-score is the metric that represents both Recall and Precision. Considering the F1-score, Neural Network has the best performance compared to all the other models, indicating

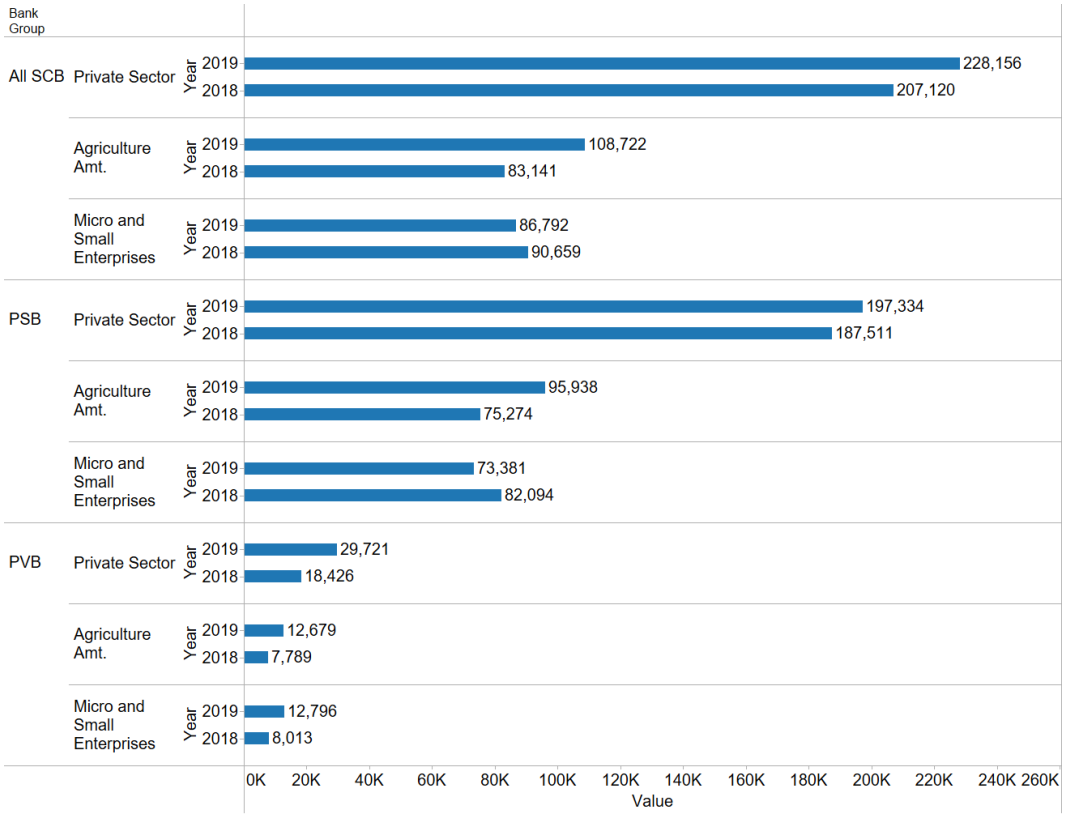

Fig. 9. Sector-wise GNPA s of Banks

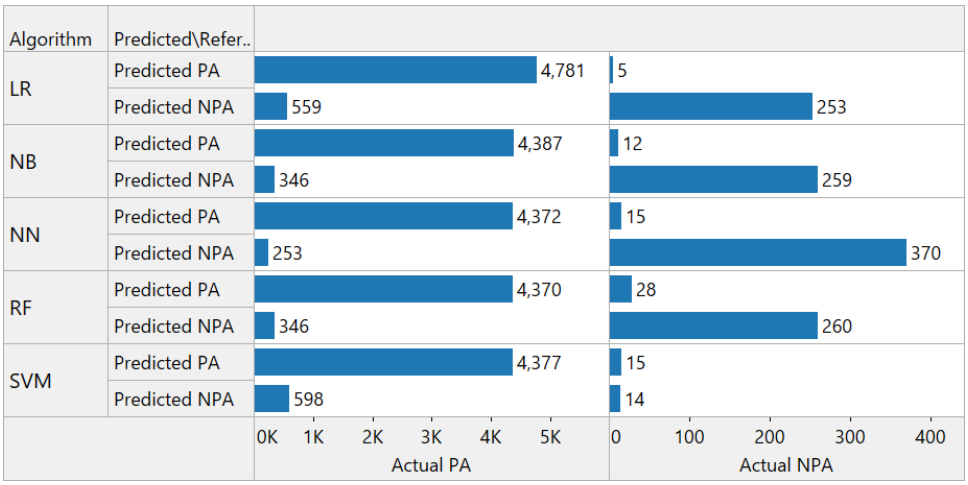

Fig. 10. Confusion matrices for classification algorithms

the least positive class misclassification. This is significant for NPA, misclassification concerning NPA class must be minimum. Specificity gives the measure of correctness of predictions concerning actual negative examples i.e the loans which are PAs. The results show that all the models have almost the same Specificity. Kappa is an important measure for evaluating a classifier's performance, especially for an imbalanced data set. It also considers the examples which are classified correctly by chance (randomly). Neural Network has the best kappa statistics indicating that it is the most suited classification algorithm for NPAs.

The calibration plot for the classification models is shown in Figure 13. It shows the agreement between the actual outcome and predictions of the model. The ideal model should have the line at 45 °. The Neural Network model is the best fit according to this plot.

The Receiver Operating Characteristic curve for all the models is shown in Figure 14. It shows the trade off between sensitivity and specificity. The model having the line near the top left corner represents the good test results. According to the ROC curve plotted Neural network has moderate performance, Random Forest and Naive Bayes show better performance, and Logistic Regression, SVM show poor performances.

Relative Commission Error (RCE) is the error that occurred due to the assignment of a sample to a class that in fact, it does not belong to. Actual positives classified as negative will lead to RCE-Positive and Actual negatives

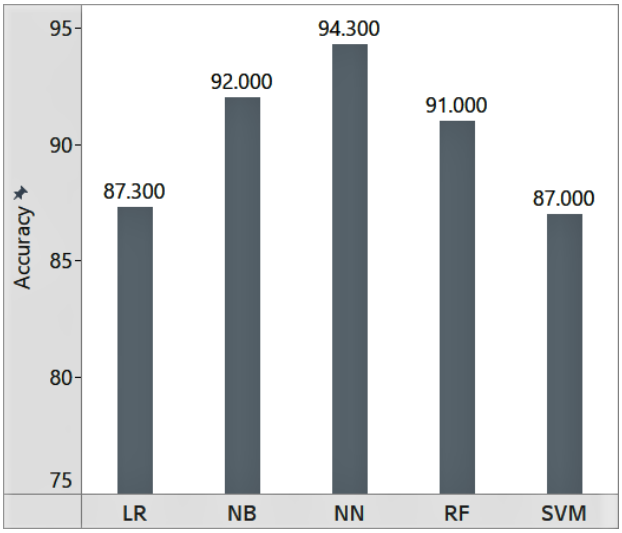

Fig. 11. Evaluation of Classification Algorithms for Prediction

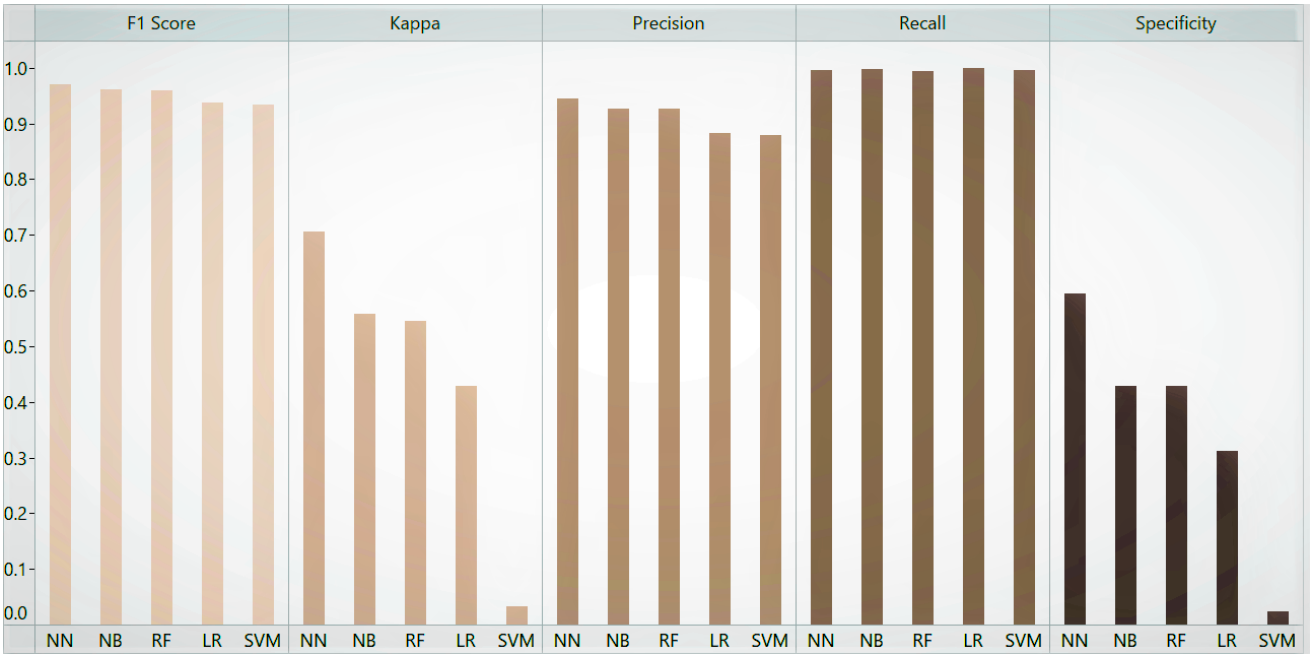

Fig. 12. Performance Metrics for Classification Algorithms

classified as positive lead to RCE-Negative. Since Negative examples are in large numbers, the error ratio is higher compared to the positive class. However, both RCE-positive and RCE-negative errors are the least in the Neural Network model.

Considering the above all metrics and graphs, it can be observed that Mr-Neural Networks performs best considering the imbalanced factor of NPA prediction, having a small number of Positive class as NPA. It is very important not to miss any NPA case by predicting it as a PA case. MR-Neural Network works well with the least misclassification error. In the given setup and chosen initial parameters for the models, MR-Neural Network is best suited for NPA prediction. Imbalancing factor can be handled by simulating samples from a class that has few samples [41]. However, since NPA detection deals with financial and behavioral data it requires expertise to validate the data.

Since the experimentations are carried out on a Hadoop cluster which is scalable, the performance of the classification algorithms considered are tested by varying the size of the cluster (incrementally adding a node) As and when the node is added, the algorithms are executed and time is recorded. The ratio between the two consecutive times is calculated which indicates the decrease in time caused due to the addition of the node. The Figure 16 shows the performance of classification algorithms on the Hadoop cluster when the number of nodes is increased linearly. It can be observed that, with the addition of a node in the cluster, the algorithms run faster with twice the speed of the previous setup. Linear regression has the highest decrease in the time when a node is added. SVM has the least reduction in time. Neural network, Naive Bayes, and Random forest show a similar decrease in time with node addition. MR-Neural Network has moderate speed and best performance

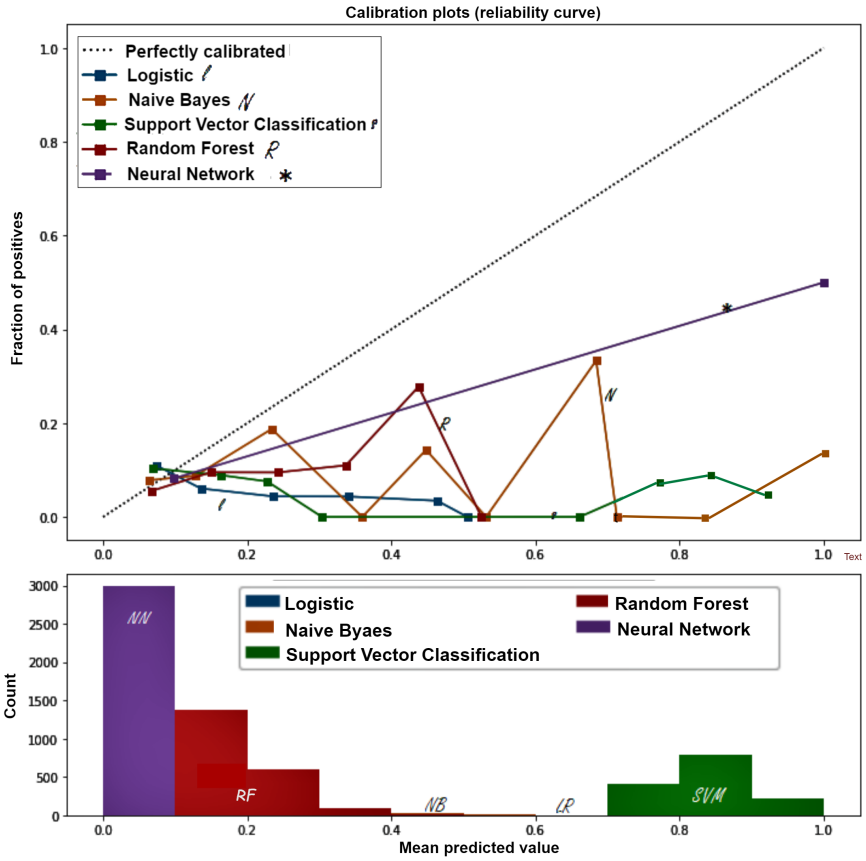

Fig. 13. Calibration plot for classification algorithms considered

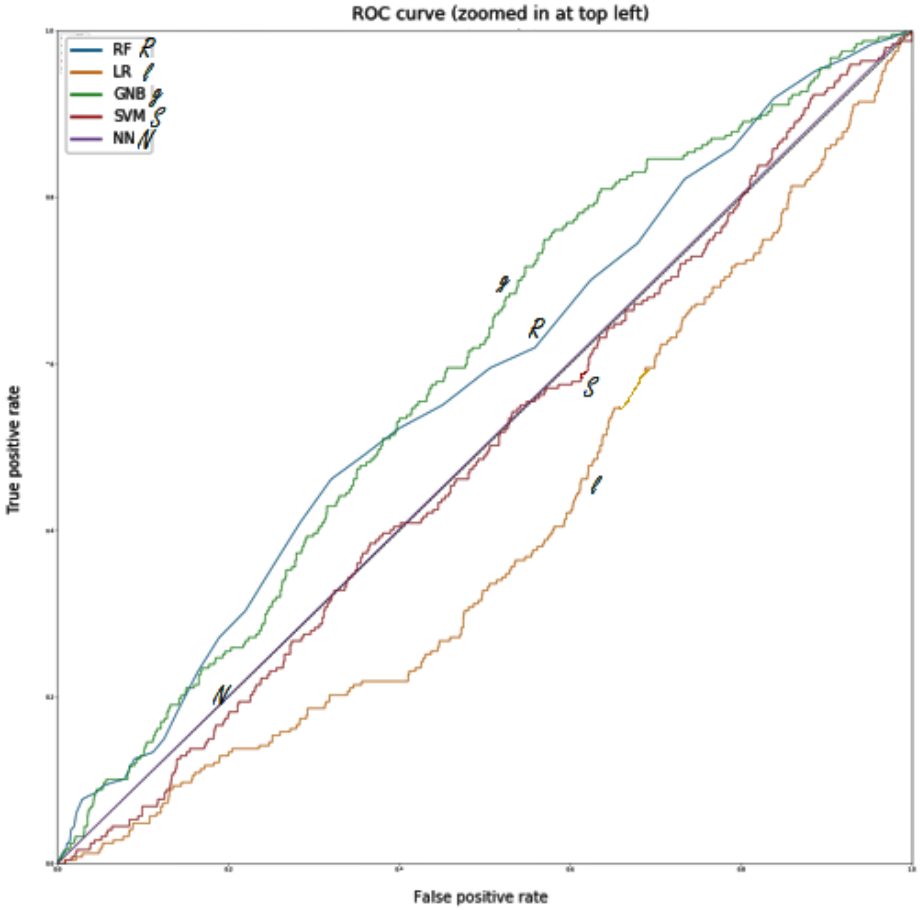

Fig. 14. ROC for Classification Algorithms

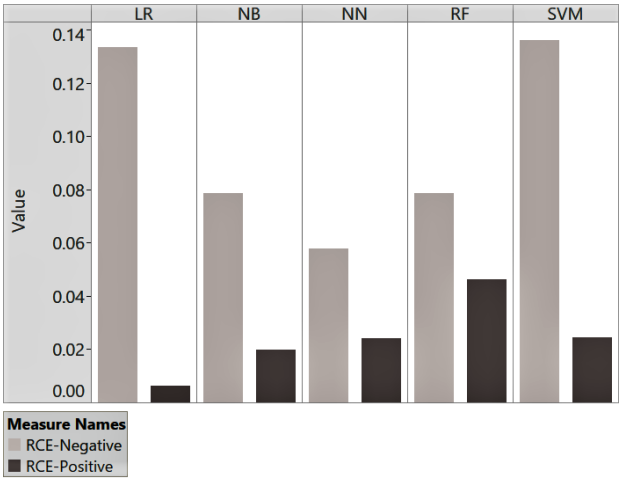

Fig. 15. Relative Commission Error for Classification Algorithms

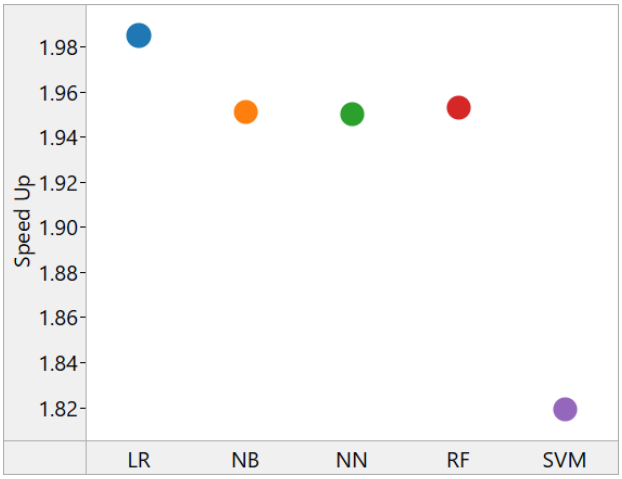

Fig. 16. Performance of Classification Algorithms on Hadoop Cluster

compared to other models, hence it can be used in the system for NPA detection. The results can be further used for analyzing whether NPA identified are wilful or genuine.

VII. Conclusion

Financial institutions face a lot of consequences due to fraudulent activities. Particularly banks face challenges when revenue-earning assets such as loan services are subjected to become non-performing wilfully. Hence NPA detection in the early stage has become a very essential requirement to the banks. NPA detection is a big data problem as models need to process both historical data and current data in real-time for early identification of NPA/Wilful default. This can be achieved by implementing the ML-based supervised models. In this paper, the Map-Reduce logics of various supervised algorithms are evolved and implemented. The algorithms are tuned and parameterized for efficient detection of NPAs for the considered loan data. The performance analysis of these algorithms shows that the MapReduce-Neural Network algorithm performs better for NPA prediction. Further, the detected/predicted NPA cases can be analyzed for wilful default using behavioral analysis and this automated process can be integrated with the existing loan management system.

References

[1] (2018, February) Unpublished Data Show India's Fraud Problems Extend Far Beyond Punjab National Bank. [Online]. Available: https://thewire.in/banking/unpublished-rbi-data-shows-bank-loan-frauds-extend-way-beyond-pnb

[2] PWC. (2018, March) PWC's Financial RegTech Insights. [Online]. Available: https://www.pwc.in/consulting/financial-services/fintech/point-of-view/financial-regulatory-technology-insights-newsletters-vinyamak/march-2018.html

[3] Deloitte. (2015, June) Deloitte India Banking Fraud Survey Edition II. [Online]. Available: https://www2.deloitte.com/content/~in/Documents/finance/in-fa-banking-fraud-survey-noexp.pdf

[4] RBI Documents. (2015, March) RBI Report of Committee on Data Standardization. [Online]. Available: https://rbi.org.in/scripts/PublicationReportDetails.aspx"

[5] Economic Times. (2019, September) 18 PSBs hit by 2,480 cases of fraud of Rs 32,000 cr in Q1: RTI. [Online]. Available: https://economictimes.indiatimes.com/industry/banking/finance/banking/18-psbs-hit-by-2480-cases-of-fraud-of-rs-32000-cr-in-q1-rti/articleshow/71036346.cms

[6] Q. Zhan and H. Yin, "A Loan Application Fraud Detection Method Based on Knowledge Graph and Neural Network," in *Proceedings of 2nd International Conference on Innovation in Artificial Intelligence*. ACM, 2018, pp. 111–115.

[7] L. H. Haß, S. Vergauwe, and Z. Zhang, "State-ownership and bank loan contracting: evidence from corporate fraud," *The European Journal of Finance*, vol. 25, no. 6, pp. 550–567, 2019, doi:10.1080/1351847X.2017.132.

[8] H. Wang, "Detection of fraudulent users in P2P financial market," in *Proceedings of 2nd International Conference on Material Engineering and Advanced Manufacturing Technology, MEAMT*. EDP Sciences, 2018, pp. 1–6.

[9] D. J. Philip, N. Sudarsanam, and B. Ravindran, "Improved Insights on Financial Health Through Partially Constrained Hidden Markov Model Clustering on Loan Repayment Data," *SIGMIS Database*, vol. 49, no. 3, pp. 98–113, 2018, doi:10.1145/3242734.3242741.

[10] A. Talavera, L. Cano, D. Paredes, and M. Chong, "Data Mining Algorithms for Risk Detection in Bank Loans," in *Proceedings of Annual International Symposium on Information Management and Big Data*, Lima, Peru, September 2018, pp. 151–159, doi:10.1007/978-3-030-11680-4_16.

[11] M. Kumar, S. Babu, and K. Prabhavathi, "Wilful default or socially responsible - A study of Indian banks and companies," in *Proceedings of International Conference on Industrial Engineering and Operations Management*. EDP Sciences, 2019, pp. 3632–3633.

[12] A. Shrivastava, L. Karthik, M. Subramanyam, and J. A R, "Prediction of Wilful Defaults: An Empirical Study from Indian Corporate Loans," *International Journal of Intelligent Technologies & Applied Statistics*, vol. 11, no. 1, pp. 1–39, 2018, doi:10.6148/IJITAS.201803_11(1).0002.

[13] K. S. Rajha, "Determinants of non-performing loans: Evidence from the Jordanian banking sector," *Journal of Finance and Bank Management*, vol. 4, no. 1, pp. 125–136, 2016, doi:10.15640/jfbm.v4n1a9.

[14] M. B. Alexandri and T. I. Santoso, "Non Performing Loan: Impact of Internal and External Factor (Evidence in Indonesia)," *International Journal of Humanities and Social Science Invention*, vol. 4, no. 1, pp. 87–91, 2015, doi:10.5539/ijbm.v9n4p22.

[15] P. K. Ozili, "How bank managers anticipate non-performing loans. Evidence from Europe, US, Asia and Africa," *Applied Finance and Accounting*, vol. 1, no. 2, pp. 73–80, 2015, doi:10.11114/afa.v1i2.880.

[16] R. A. Rachman, Y. B. Kadarusman, K. Anggriono, and R. Setiadi, "Bank-specific Factors Affecting Non-performing Loans in Developing Countries: Case Study of Indonesia," *The Journal of Asian Finance, Economics and Business*, vol. 5, pp. 35–42, 2018, doi:10.13106/jafeb.2018.vol5.no2.35.

[17] W. Anjom and A. M. Karim, "Relationship between non-performing loans and macroeconomic factors with bank specific factors: a case study on loan portfolios–SAARC countries perspective," *ELK Asia Pacific Journal of Finance and Risk Management*, vol. 7, no. 2, pp. 1–29, 2016, doi:10.16962/EAPJFRM/issn.2349-2325/2015.

[18] S. Dhar and A. Bakshi, "Determinants of loan losses of Indian Banks: a panel study," *Journal of Asia Business Studies*, vol. 9, no. 1, pp. 17–32, 2015, doi:10.1108/JABS-04-2012-0017.

[19] J. K. Bawa, V. Goyal, S. Mitra, and S. Basu, "An analysis of NPAs of Indian banks: Using a comprehensive framework of 31 financial ratios," *IIMB Management Review*, vol. 31, no. 1, pp. 51 – 62, 2019, doi:10.1016/j.iimb.2018.08.004.

[20] P. Kanhere and H. K. Khanuja, "A Survey on Outlier Detection in Financial Transactions," *International Journal of Computer Applications*, vol. 108, no. 17, pp. 23–25, Dec 2014, doi:10.5120/19004-0502.

[21] A. Maniyar and P. Dandannavar, "Outlier Detection In Financial Transactions: A Review," *International Journal of Emerging Technology in Computer Science & Electronics*, vol. 14, no. 2, pp. 412–420, 2015, doi:10.1016/S1473-3099(14)00424-2.

[22] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, no. 1, pp. 406–421, 2018, doi: 10.1016/j.patcog.2017.09.037.

[23] D. A. A. G. Singh and E. J. Leavline, "Model-Based Outlier Detection System with Statistical Preprocessing," *Journal of Modern Applied Statistical Methods*, vol. 15, no. 1, pp. 789–801, 2016, doi:10.22237/jmasm/1462077480.

[24] S. N. John, C. Anele, O. O. Kennedy, F. Olajide, and C. G. Kennedy, "Realtime Fraud Detection in the Banking Sector Using Data Mining Techniques/Algorithm," in *Proceedings of International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2016, pp. 1186–1191.

[25] M. Sanchez, J. Torres, P. Zambrano, and P. Flores, "FraudFind: Financial fraud detection by analyzing human behavior," in *Proceedings of 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018, pp. 281–286.

[26] R. Laimek, N. Kaothanthong, and T. Supnithi, "ATM Fraud Detection Using Outlier Detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics.)*, vol. 11314, no. 1, pp. 539–547, 2018, doi:10.1007/978-3-030-03493-1_56.

[27] N. Malini and M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection," in *Proceedings of 3rd IEEE International Conference on Advances in Electrical and Electronics, Information, Communication and Bio-Informatics*. IEEE, 2017, pp. 255–258.

[28] P. Femi and S. Ganesh Vaidyanathan, "Comparative Study of Outlier Detection Approaches," in *Proceedings of International Conference on Inventive Research in Computing Applications*. IEEE, 2018, pp. 366–371.

[29] Carcillo, F. and Le Borgne, Y.-A. and Caelen, O. and Kessaci, Y. and Oblé, F. and Bontempi, G., "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 1, no. 1, pp. 1–15, 2019, doi:10.1016/j.ins.2019.05.042.

[30] L. Wee-Yong, A. Sachan, and V. Thing, "Conditional weighted transaction aggregation for credit card fraud detection," in *Proceedings of International Federation for Information Processing 2014*, 2014, p. 3–16.

[31] M. Sathyapriya and V. Thiagarasu, "A Cluster based Approach for Credit Card Fraud Detection System using HMM with the Implementation of Big Data Technology," *International Journal of Applied Engineering Research*, vol. 14, no. 2, pp. 393–396, 2019, doi:10.1109/WICT.2011.6141395.

[32] J. J. Xu, L. Yong, and C. Michael, "P2P Lending Fraud Detection: A Big Data Approach," in *Intelligence and Security Informatics*. Springer, 2015, pp. 71–81, doi:10.1007/978-3-319-18455-5_5.

[33] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, ""A survey of open source tools for machine learning with big data in the Hadoop ecosystem"," *Journal of Big Data*, vol. 2, no. 1, pp. 1–24, 2015, doi:10.1186/s40537-015-0032-1.

[34] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big Data Clustering: A Review," in *Proceedings of International Conference on Computational Science and Its Applications*. Springer, 2014, pp. 707–720.

[35] S. Zheng, *Naïve Bayes Classifier: A MapReduce Approach*, 1st ed. North Dakota State University, 2014.

[36] J. Bell, *Machine Learning for Big Data: Hands-On for Developers and Technical Professionals*, 1st ed. Wiley, 2014.

[37] F. Ö. Çatak and M. E. Balaban, "A MapReduce-based distributed SVM algorithm for binary classification," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 24, no. 3, pp. 863–873, 2016, doi:10.3906/elk-1302-68.

[38] J. Z. Lei and A. A. Ghorbani, "Improved competitive learning neural networks for network intrusion and fraud detection," *Neurocomputing*, vol. 75, no. 1, pp. 135 – 145, 2012, doi:10.1016/j.eswa.2013.05.021.

[39] Y. Sahin, S. Bulkan, and E. Duman, "A Cost-ensitive Decision Tree Approach for Fraud Detection," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916 – 5923, 2013, doi:10.1016/j.eswa.2013.05.021.

[40] RBI, "Operations and performance of commercial banks," December.

[41] C. Wang and Y. Yang, "Nearest Neighbor with Double Neighborhoods Algorithm for Imbalanced Classification," *IAENG International Journal of Applied Mathematics*, vol. 50, no. 1, pp. 147–159, 2020.