The Effect of Imbalanced Data and Parameter Selection via Genetic Algorithm Long Short-Term Memory (LSTM) for Financial Distress Prediction

Juliana Adeola Adisa, Samuel Ojo, Pius Adewale Owolawi, Agnieta Pretorius, Sunday Olusegun Ojo,

Abstract—**Financial companies are grappling with a burning issue about bankruptcy prediction. There are many methods for bankruptcy prediction, including statistical models and machine learning. Real-life datasets are often imbalanced with high dimensionality. Therefore, it is challenging to train a robust model to predict bankruptcy. Thus, we first applied an oversampling technique known as the Synthetic Minority Oversampling Technique (SMOTE) to reduce the skewness of the data. The balanced data was trained with the baseline models, the ensemble classifiers using different combination methods and the long short-term memory (LSTM) model. In addition, we employed an optimization technique called a genetic algorithm (GA) to optimize and determine the learning parameters of an LSTM network. We further determine the effects of using different training/testing ratios on the developed models. An autoencoder long short-term memory (LSTM) model was developed to extract the best feature representation of the input data. A comparative analysis was carried out between the LSTM-GA and autoencoder-LSTM. The results show that the improved LSTM-GA model with an accuracy of 98.11% performs better than other models. Overall, the research work concluded that all models and LSTM have good performances, while the optimized LSTM model via genetic algorithm outperforms the classical machine learning models.**

Index Terms—**genetic algorithm, long short-term memory networks,Parameter Selection, financial distress prediction, bankruptcy prediction.**

I. Introduction

BANKRUPTCY is an undesirable event. The negative impact of bankruptcy on the economy last for a very long period, and the effects can be felt by business owners, shareholders, investors, policymakers, employees, and the government [1]. The Lehman brother's financial crisis of 2008 affected the world, which has led to an increase in the intensity of research towards the development of new architectures for crisis prediction and management [2]. Therefore, bankruptcy prediction plays an important role in financial analysis because of its significant impact on economic decisions. It helps preventive measures well in advance, giving concerned people ample time to act. Researchers around the globe are paying attention to this issue, and the study of bankruptcy prediction has greatly increased over the past few decades [1], [3], [4]. Many factors could lead to financial distress, including interest rate volatility, excessive risk-taking, inadequate internal control mechanisms, and poor management practices [5]. Financial distress (FD) does not occur abruptly but takes gradual evolvement; hence, it can be predicted.

According to research studies, machine learning approaches include support vector machines (SVM), artificial neural networks (ANN), decision trees (DT) and deep-learning methods like LSTM outperform statistical methods [6], [7]. In specific, the combination of multiple machine learning classification techniques to form an ensemble model performs better than a single ML classification technique [8], [9]. The superiority of the ensemble methods over single classification models has been demonstrated in many related works of literature [6], [9]. However, there is a need to point out that the performances of ensemble classifiers are usually domain-dependent [8].

Financial distress is a cumulative event because it is unlikely that financial crises will occur in a short-term economic context. Therefore, a learning model for financial prediction must capture long-term economic dependencies. Hence, the introduction of LSTM for bankruptcy prediction. LSTM employed the memory modules to solve long-term dependency, vanishing gradient, or exploding gradient problems [10], [11]. In addition, the performance of LSTM has been further improved through the application of various optimization techniques [12], [13].

As opposed to traditional non-linear optimisation techniques, a genetic algorithm (GA) searches through a population of possible solutions to generate an improved solution rather than incrementally changing one solution at a time [14]. Researchers have employed GA in many applications, and results have proven that GAs are effective and robust in searching large spaces during their application to different fields [12], [15]. Genetic algorithms have been employed to optimise LSTM for stock market prediction [12], [15], [16]. GA was used to select the LSTM window size in stock market prediction [12], while other studies which carried out bankruptcy prediction used GA as a stand-alone model [13].

Hybrid machine learning models have shown better per-

Manuscript received April 19, 2022; revised March 15, 2023.

This work is supported by National Research Foundation (NRF) South Africa, Grant number 12908. The authors acknowledge the Computer Systems Engineering department of the Tshwane University of Technology for providing the facilities used for this research.

Juliana Adeola Adisa is a PhD Student at Tshwane University of Technology, Department of Computer Systems Engineering, Soshanguve Campus, South Africa (corresponding phone: +27848845930; e-mail: adeolaa33@gmail.com).

Samuel Ojo is a PhD Student at Tshwane University of Technology, Department of Computer Systems Engineering, Soshanguve Campus, South Africa (e-mail: 208322842@tut4life.ac.za).

Pius Adewale Owolawi is the Head of Department of Computer Systems Engineering, Tshwane University of Technology, Soshanguve Campus, South Africa (e-mail: owolawipa@tut.ac.za).

Anieta Pretorius is the Deputy Dean Faculty of ICT, Tshwane University of Technology, Soshanguve Campus, South Africa (e-mail: pretoriusab1@tut.ac.za).

Sunday Olusegun Ojo is a Professor of Computer Science, Department of Information Technology, Durban University of Technology, Durban, South Africa (e-mail: sundayO1@dut.ac.za).

formance than stand-alone models [17], [18]. Hence, this study deploys hybrid GA-LSTM in the prediction of financial distress. Specifically, our study employed GA to find the optimal LSTM architectural structure for financial distress prediction. Based on the background, this study will develop a method for bankruptcy prediction that uses a long short-term memory (LSTM) layer. Specifically, the aim is to develop an improved LSTM for bankruptcy prediction by employing a genetic algorithm, an optimisation method that will determine the best parameters for the LSTM algorithm. The research will compare the improved LSTM to other machine learning models. Section II presents the background of the proposed framework. In Section III, the experimental setup and methodology are presented. Section IV discusses the experimental results, and Section V presents the conclusion.

Bankruptcy prediction requires inferring meaningful insight from a large amount of historical data. The financial state of firms is usually described mathematically through different indicators [2]. The availability of the data is helpful in bankruptcy prediction. However, the class imbalance problem and high dimensionality of the dataset make accurate prediction difficult, despite the availability of the data [19], [20], [21]. The main contributions include the following:

1) Application of SMOTE and principal component analysis (PCA) for data imbalance and high dimensionality: This will resolve the class imbalance problem and high dimensionality associated with the bankruptcy data.

2) Application of genetic algorithm (GA): The optimal parameters will be selected for our LSTM model through the application of GA.

3) Applied LSTM model: This resolves the vanishing and exploding gradient problem of long-term dependencies.

II. Theory

Several authors have employed single classification methods in financial distress prediction. The most frequently used supervised machine learning methods are DT, SVMs and MLP-ANN [8]. This section will introduce these techniques as our baseline models and briefly explain the concept of recurrent neural networks, the variant called LSTM, and the experimental setup.

A. Artificial Neural Networks (ANNs)

ANN is an appropriate model for predicting and dealing with a large dataset volume in data mining because meaningful information can be extracted from a large volume of data [22]. ANN is a black box non-parametric classifier that does not need assumptions about the distribution densities [23]. The ANN create an architecture that connects neurons among layers [24]; the sets of input variables are mapped to the output variables. In this study, we applied multilayer perceptron (MLP), which is composed of three layers: an input layer, a hidden layer, and an output layer. Every neuron in the input and output layers is connected to the hidden layer neurons. The MLP employed the back-propagation learning technique to estimate the learning weights and minimize the classification error. The equation of a neural network is given by [25]:

$$w_i = \sum_j w_{ij} x_j + Q_j \tag{1}$$

$$y = f(w_i) \tag{2}$$

The output value of a layer is represented by y, w_i denotes the activation value of the ith node in one layer, while x_i is the input signal, the bias is expressed as Q_j, and w_ij is the weight that connects two nodes. The error is calculated at the output layer. The formula given is employed to adjust the weight as follows:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\delta E}{\delta w_{ij}} \tag{3}$$

The error denoted by E is the difference between the predicted output and expected output, and δ represents the learning rate.

B. Support Vector Machines

Support Vector Machine (SVM) is a non-discriminative learning model. SVM is an efficient solution to linear, non-linear classification and regression. SVMs are non-probabilistic and designed to use hyperplane boundaries to divide the training vectors into categories of different data patterns. SVM carry out a classification task by constructing an optimum hyperplane with the maximum distance between the different class labels' data points. Some data points called support vectors (SVs) surround the hyperplane. The SVs influenced the position and orientation of the hyperplane. SVs help to maximize the margin of the classifier. However, deletion of the SVs means a change in the hyperplane position. The optimal hyperplane that separates the data into two different classes is a linear function. The hyperplane equation given by [26],

$$f(x) = w|x + b = \sum_{i=1}^{N} (w_i x_i) + b = 0 \tag{4}$$

Extension of SVM to non-linear classification entails using kernel trick to transform a non-linear classification task into a linear problem [27]. Non-linear data are converted to high dimensional space because they are widely scattered and easy to separate. Research proved that "for any data set, there exists a kernel function that will allow the data to be linearly separated" [27]. The kernel Φ function is the inner dot product of the new vectors. The most common kernel functions are linear, polynomial, Gaussian radial basis function and the sigmoid. Their respective equations are as follows: deg is the degree of the polynomial, C is the regularisation parameters, and gamma γ is an adjustable parameter of the kernel function.

C. Decision Tree

A decision tree is a technique for classification or prediction purposes. DTs have a flowchart structure. They used a "divide-and-conquer" approach to splits the data to create leaves. DT is built such that the internal node shows the testing on the attribute. Each branch denotes the testing outcomes, and the class label represents the leaf nodes, the resolution taken after all the features were computed. In DT, the node with the highest information gain is the root node. It compares all the features until all the features contrast when there are no more attributes for further partitioning [28].

D. Long Short Term Memory Networks

Long-Short Term Memory networks (LSTMs) are variants of RNNs that solve the long-term dependencies learning problem; LSTMs are used for modelling sequence learning. The significant difference between LSTMs and RNNs is that each repeating module in LSTM is composed of four connected layers. An LSTM module can effectively mitigate long-term dependence and retain useful information about sequence data by controlling the gates [29]. There are four gates in an LSTM model developed out of a sigmoid and a pointwise multiplication operation. These gates are optionally used to let information through, and their equations are stated as given by [11]:

Forget gate removed the unwanted information from the cell state,

$$f^t = \sigma(W_f[o^{(t-1)}, x^t] + b_f) \tag{5}$$

The input gate determines the new information that needs to be added to the cell state,

$$i^t = \sigma(W_i[o^{(t-1)}, x^t] + b_i) \tag{6}$$

$$\hat{C^t} = tanh(W_c[o^{(t-1)}, x^t] + b_c) \tag{7}$$

The old cell state is updated $C^(t-1)$ to a new cell state C^t,

$$c^t = f^t * c^{(t-1)} + i^t * \hat{C}^t \tag{8}$$

Output gate determine the output of the LSTM;

$$h^t = \sigma(W_h[o^{(t-1)}, x^t] + b_h) \tag{9}$$

$$o^t = h^t * tanh(c^t) \tag{10}$$

$$y^t = \sigma(W^{out} o^t + b^{out}) \tag{11}$$

The Forget Gate is the gate that identifies the information that needs to be removed from the memory cell state.

$$f^t = \sigma(W_f[o^{(t-1)}, x^t] + b_f) \tag{12}$$

E. Genetic Algorithm (GA)

GA is an optimization procedure that uses a stochastic search heuristic method employed to search a complex space, mimicking the natural evolution process as modelled by the Darwinian evolution [14]. GA belongs to the evolutionary algorithms group that employs natural selection to approximate solutions for a given problem. The operations of the GA include selection, crossover and mutation.

The fitness function is used to assess the quality of a solution [30]. The fitness function for each member of the population is calculated individually, and the highest value is chosen. Subsequent generations of solutions are generated based on the values of the fitness functions. Algorithm performance relies heavily on fitness functions [30]. The search for a solution may fail if the fitness function is poor.

Crossover: New individuals are created by crossing the selected parents. During the selection process, genomes are randomly selected from the parents. At a random point, the genome of one parent is switched with the genome of the other parent. It is called a single-point crossover.

Mutation: This allows new solutions to be discovered. At a certain probability, the bits within the genome are altered, and mutation takes place. In other words, the genome is replaced with random bits. Mutation creates an individual from just one parent and increases genetic diversity. It prevents individuals from having two identical genomes. Hence, mutation is the process of preventing evolutionary stagnation at a local minimum.

F. Combination Methods

The integration of multiple classification decisions brings forth a model with reliable and more accurate performance. This integration is called an ensemble method, and research studies show that ensemble techniques are superior to individual classification methods [31]. There are three different approaches to ensemble methods, and we will discuss them in this section. Bagging: This is otherwise called bootstrap aggregating. In bagging, training multiple models of the same learning algorithms occurs with subsets of a randomly selected replaceable data set. Models are built on each sample (parallel). The various models vote to approve the final model and make predictions. Boosting: This is like bagging, but with slight variation. The training of each classifier is dependently employing a different training set. Models are built-in series, and it uses the learning from the previous model to adjust the weight in each successive model.

G. Autoencoder

Autoencoders are essentially non-recurrent feedforward neural networks that resemble multilayer perceptrons. Autoencoders are used to encode the training data [32]. The learning phase of an autoencoder includes the reduction of the reconstruction error without emphasis on discrimination [33]. It consists of two components: an encoder and a decoder consisting of a hidden layer, an input layer, and an output layer. The input dimension and output dimension of an autoencoder are the same. The input dimension is transformed into a hidden representation, which has a different dimension than the input and output dimensions. The hidden representation is used to reconstruct the input features basically for dimension reduction [34]. An autoencoder network is trained using samples to optimize reconstruction error. The autoencoder cost function is given:

$$T_{auto} = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\left\|\hat{x}_l - x_i\right\|^2) \tag{13}$$

where x_i and $\hat{x}_l$ are the input and output features respectively. The number of input samples is denoted by n.

H. Imbalanced data and High Dimensionality

In binary classification problems, data imbalances can occur when a small percentage of the population shows a positive result. Undersampling or oversampling the class during training are the general ways to handle imbalanced data at the data level [35]. Undersampling and oversampling techniques add bias to the data set [36]. This study employed Synthetic Minority Oversampling Technique (SMOTE) to get the best results. SMOTE is a way to generate synthetic observations of the minority class instead of oversampling with replacement. The synthetic data are generated based on similarities between existing minority examples in the feature space instead of the data space [36]. Synthetic observations are generated along the line segments joining a portion or all the k-nearest minority neighbours based on a given minority class observation [36]. SMOTE requires five k-nearest neighbours. A random neighbour is selected from the K nearest neighbours based on the amount of oversampling required.

Lean at al. [37] proposed a high-dimensionality-trait-driven learning model for feature extraction and classifier selection to improve accuracy and solve the high-dimensionality issue in credit risk assessment. The results show an improved accuracy in the proposed model for handling the dimensionality problem compared to the benchmark models listed in the study. The high dimensionality of data has been addressed in some research papers through the application of principal component analysis (PCA) [38], [39]. Building models using PCA has become more time-efficient due to the removal of noisy and redundant features.

III. Experimental Setup

This section presents and explains the experimental setup for the proposed framework, the metrics employed for performance evaluation, and the data set.

A. Data set Analysis

Specifically, Table 1 summarizes the data set based on the total instances, the number of attributes, the number of minority cases, the number of majority cases, and the imbalance ratio (IR). UC Irvine Machine repository provides access to the data set.

The research employed the Polish bankruptcy data; it is available on the repository with the link https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data#. There are 1000 Polish companies where 19.4% were bankrupt between 2000-2012, and those in operation were evaluated from 2007 to 2013. The first year of Polish Data has 271 bankrupted companies and 6756 firms still operating in the forecasting period. The total number of observations for this period is 7027 instances. The polish data set is imbalanced; that is, the number of observations in each class is not represented equally (271: 6756, 4% to 96%). Hence, classifiers have low prediction accuracy for the minority class, and new samples are classified in the majority class. The classifier's performance assessment is critical when the data is imbalanced because model performances depend on AUC [40]. First, the SMOTE was applied to generate synthetic data. The application of SMOTE randomly creates a sample of the attributes from observation in the minority class samples. SMOTE [41], [42] is used to add copies of instances from an under-represented class. The Polish data set number increased to 11092 (6756: 4336, 60.9% to 39.1%).

TABLE I
Dataset Information

Dataset	Total Instances	No of Attributes	Minority Class	Majority Class	IR
Polish Data	7027	65	271	6756	0.040
Polish + SMOTE	11092	65	4336	6756	0.64

B. Evaluation Metrics

The accuracy (Acc) and error rate are the two most frequently used metrics to assess the performance of various classification algorithms. However, imbalanced data is influenced by its distribution. Therefore, the accuracy is skewed in favour of the majority class. Additional metrics are incorporated to account for the imbalanced nature of the employed data set, such as recall, F-measures, AUC, precision, g-means, and Kappa statistics. The speed of computation and convergence rate of the iterations were calculated using the computation time (CT) measured in seconds. Table II gives the mathematical equations for the evaluation metrics.

Accuracy (ACC) measures the proportion of correct points over the total points. Recall or sensitivity measures correctness taken from the number of successful, positive cases (TP); the percentage of correctly classified bankrupt samples.

Precision: It is a measure of exactness that determines the number of instances classified as true positive from the total classified instances. It gives information concerning false positives. A low precision means a large number of false positives [43].

Specificity: It is the opposite of recall. It measures the frequency of correctly classified negative incidents to the total number of all negative instances. It refers to how often the negative incidents are classified correctly.

A Receiver Operating Characteristic curve (AUC): The ROC curve measures the probability of class separation, while the AUC curve measures the degree of separation and presents a trade-off between a true positive and a false positive [23]. The model with a higher AUC has a better performance at distinguishing between classes.

In binary classification, the F-measure measures the average of precision and recall. Where σ represents the sigma. A geometric mean (G-mean) examines the degree of bias as expressed by comparing the accuracy of positive and negative classes. Classifiers with low G-means are biased in favour of one of the classes.

Kappa statistic gives the quantitative measure of the true agreement in any situation beyond what could be achieved by chance. In binary classification problems, kappa statistic is used to report how well two observations agree. The kappa statistic is expressed in Table II, where Po represents the observed agreement, and Pe denotes the expected agreement obtained by chance. Landis and Koch [44] provide a common scale for interpreting kappa results, which values ranging from 0 to 1. In the case of a kappa value of 1, the observations are perfectly in agreement, while a kappa value of 0 signifies complete disagreement between the observations [44].

$$Kappa = \frac{P_o - P_e}{1 - P_e} \tag{14}$$

Classification quality can also be determined by the Matthews correlation coefficient (MCC) and its values between -1 and +1. The closer the MCC gets to +1, the more accurate the classification task becomes [45]. The mathematical formula for each performance metric is given in Table II:

TABLE II
PERFORMANCE METRICS

Accuracy	Error	RMSE	Precision
$\frac{TP+TN}{TP+TN+FP+FN}$	1 - Accuracy $= \frac{FP+FN}{TP+TN+FP+FN}$	$\sqrt{\frac{\sum_{k=1}^{N}(y_k - \hat{y}_k^t)^2}{N}}$	$\frac{TP}{TP+FP}$

Recall/Sensitivity	F-Measure	Geometric mean	Specivicity
$\frac{TP}{TP+FN}$	$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	$\sqrt{\text{sensitivity} \times \text{Specificity}}$	$\frac{TN}{TN+FP}$

C. Proposed Framework

The experimental setup in Fig. 1 shows all the stages involved in the proposed framework of this research. It started with the preliminary model development, such as data preprocessing, normalization, and removal of outliers. Followed by the actual model development includes artificial neural network (ANN), support vector machine (SVM), decision tree (DT), LSTM optimization and comparison to ensemble classifiers. Lastly, the post-model development deals with the analysis of the results. The proposed framework was developed using the python language and different packages such as Bernoulli, bitstring, rcParams, and arff. The framework in this research combines SMOTE algorithm and parameter tuning created and the packages in python to overcome the class imbalance of the Polish bankruptcy data set and remove outliers. Statistics and data analysis using python as the programming language.

We applied the SMOTE to the data set and investigated the effect of different data division ratios on the baseline models (ANN, SVM, DT). Additional two data balancing techniques were applied to compare with the proposed SMOTE application method: adaptive synthetic (ADASYN) and upsample minority class. The ensembles were developed from the baseline models using bagging and boosting as the combination methods. The performances were evaluated based on relevant metrics such as accuracy, AUC, computation time, and others, as stated in section III.B. The ensemble classifiers were developed with ten fixed numbers of classifiers. Table III shows the learning parameters used for this research work. The single classifiers, ensemble classifiers and the LSTM learning parameters are all given in Table III. A batch size of 100 was used for all the models.

TABLE III
MODELS LEARNING PARAMETERS

Models	Parameters		
DT	pruned = yes	Minimum of instances per leave = 2	Confidence factor = 0.25
SVM	Kernel = Linear	C=5	Gamma = 0.0
MLP	Hidden layer = 1	Learning rate = 0.3	Activation Function =Relu
Bagging	Classifiers = DT, SVM, MLP	No of combined classifiers = 10	Combination method =Bagging
Boosting	Classifiers = DT, SVM, MLP	No of combined classifiers = 10	Combination method =Boosting
Stacking	three base learners (DT, SVM, MLP)	metaclassifier = LSTM	Activation Function = Relu
LSTM	Activation Function = Relu	Gate Activation function = Sigmoid	Output layer = Softmax

The LSTM network was optimised by a genetic algorithm (GA) to develop the LSTM-GA model. As described above in the literature review, the optimization algorithm search called GA was used to search the LSTM network and determine the best parameters for the LSTM architecture. The parameters include epoch number, batch size, number of LSTM units, learning rate, and dropout. The fitness function needs to be carefully selected. Hence, the fitness function was computed for each chromosome using the mean square error (MSE), and the set of architectural factors with the smallest MSE is the best solution. The chromosome size, population size, crossover rate and mutation rate are given as 10, 50, 0.6, and 0.003, respectively. An autoencoder LSTM was introduced for comparative analysis with the proposed model (LSTM-GA). The data were subjected to all stages before the autoencoder was applied for the feature selection process.

IV. RESULTS ANALYSIS AND DISCUSSION

In this section, we present the experimental results obtained from the research study. Each table demonstrates the performance recorded based on accuracy, precision, AUC, sensitivity, specificity, F-measure, G-mean, computation time, and kappa statistic value. The framework was applied to the Polish bankruptcy data set. The classification algorithms were applied to the balanced data set. Table IV shows the experimental results for the balanced Polish bankruptcy data set, and each method is compared to its performance against different evaluation metrics indicated in the table.

A. Models Results

Table IV shows the recorded values for all the performance metrics. The first column represents the different models employed in this research, while columns 2 to 11 are the metrics described in section III.B. The computation time (seconds) measured the time taken by each model to complete the classification task. DT ensembles have the least computation time, while it takes several minutes for SVM and MLP ensembles to finish the computation. The LSTM algorithm takes hours to compute the models. The LSTM has the best performance in terms of accuracy, followed by DT-boosting and DT-bagging.

Figure 2 shows the accuracy of the baseline model DT and the ensembles (boosting and bagging) models at different training/testing ratios. The J48 ensembles have better performance than their baseline model. J48-boosting at a 70/30 training/testing ratio has the best performance with an accuracy of 97.51%.

Figure 3 and Figure 4 show SVM, MLP and their ensembles at different training/testing ratios, respectively. The results show that MLP-bagging (10-fold cross-validation) performs better with an accuracy of 92.25%. MLP-boosting at 10-fold cross-validation followed with an accuracy of 92.09% and MLP-bagging at a 50/50 training/testing ratio with an accuracy of 91.90%.

B. LSTM-Autoencoder Model

It is noteworthy that LSTM-Autoencoder performs reasonably well, as shown in Table V. The accuracy of the LSTM-Autoencoder increases as the threshold increases. The best performance with 94.3% accuracy was achieved at the

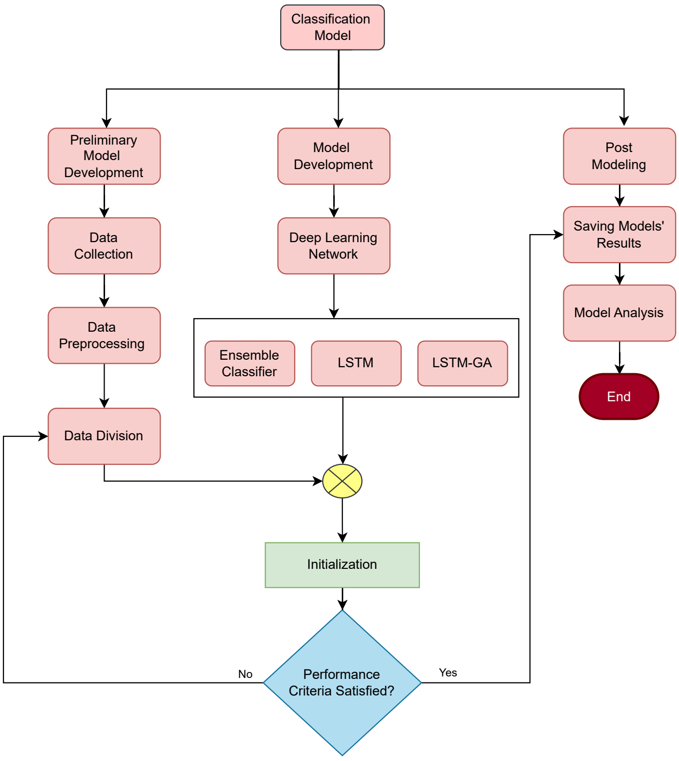

Fig. 1. Methodology Framework

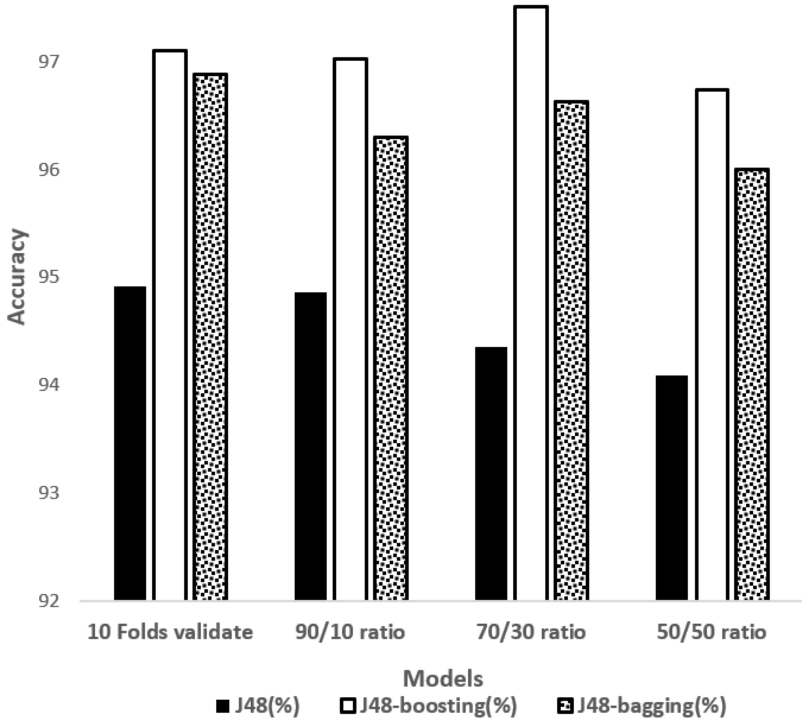

Fig. 2. DT-Ensembles at different training/test ratio. The result shows that J48-boosting perform better than others at different training/testing ratio.

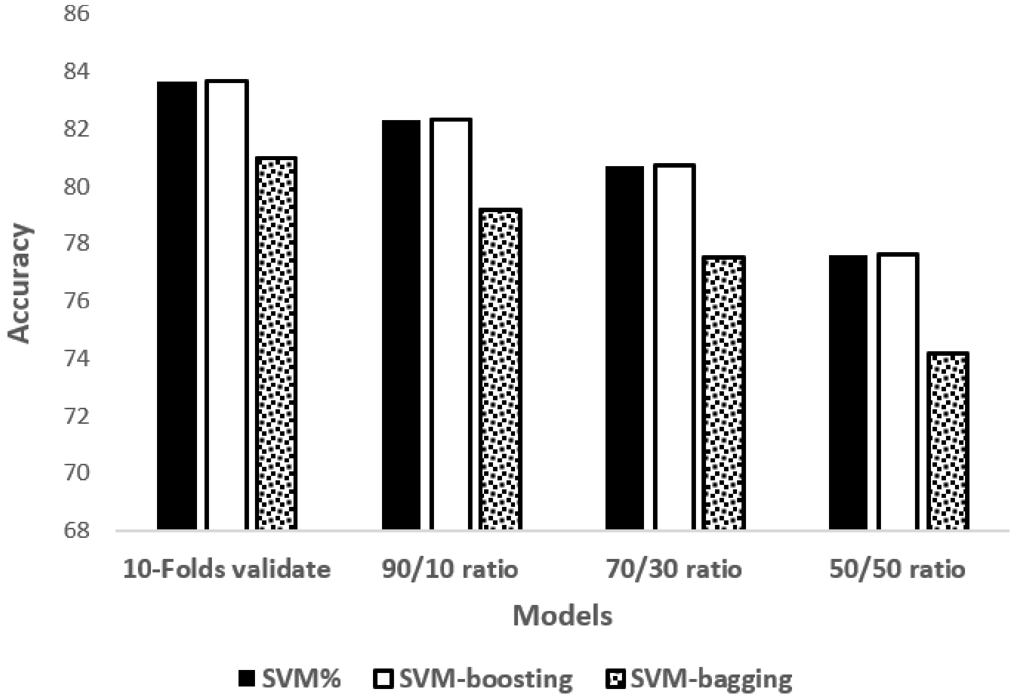

Fig. 3. SVM-Ensembles at different training/test ratio. 10-folds cross-validation is the best model for SVM-ensembles.

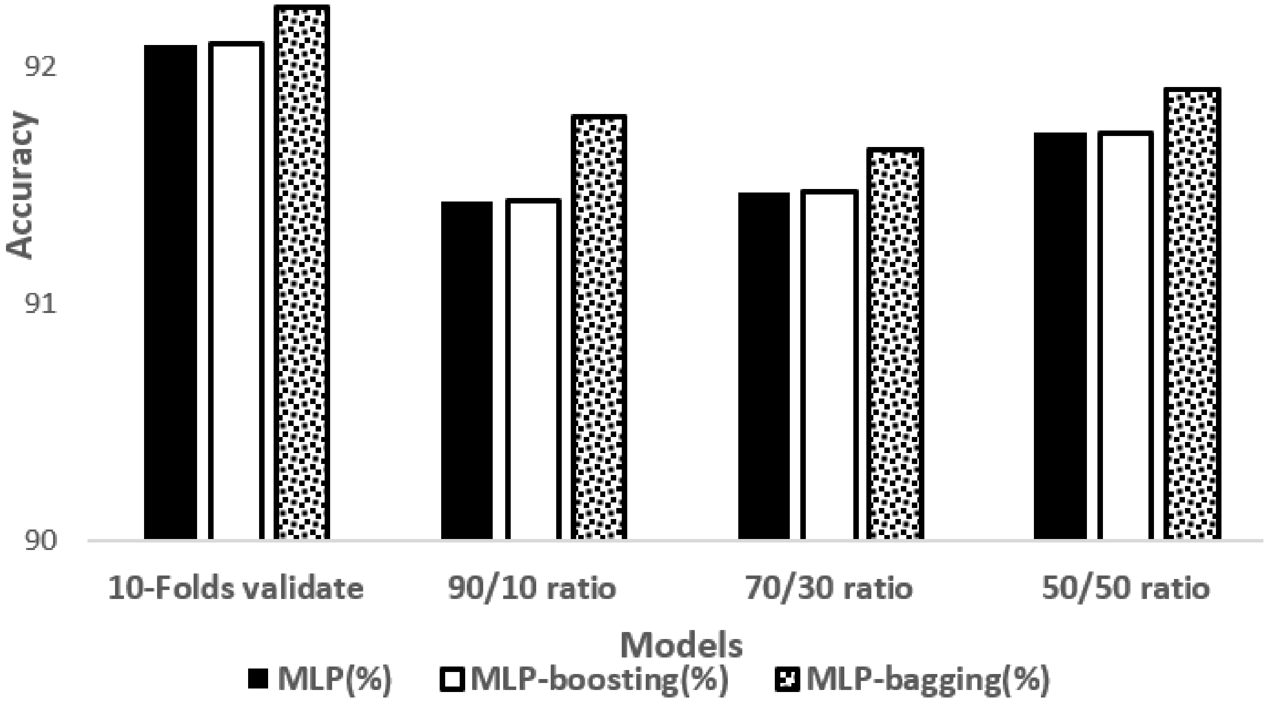

Fig. 4. MLP-Ensembles at different training/test ratio. The result shows that at 10-folds cross-validation, the MLP-ensembles perform better than other training/ testing splitting ratio.

TABLE IV
COMPARISON OF MODELS' RESULTS

Models	Accuracy	Precision	AUC	Sensitivity	Specivicity	F-measure	G-Mean	Computation Time	Kappa St.
DT	94.91	0.949	0.955	0.949	0.926	0.949	0.937	1.96	0.8929
SVM	77.62	0.836	0.716	0.776	0.656	0.749	0.713	763.02	0.629
MLP	91.43	0.922	0.957	0.914	0.804	0.913	0.857	246.86	0.8177
DT-boosting	97.01	0.971	0.990	0.971	0.951	0.971	0.960	41.65	0.9388
SVM-boosting	83.65	0.871	0.791	0.837	0.745	0.810	0.784	27650.57	0.4756
MLP-boosting	92.09	0.927	0.924	0.921	0.813	0.919	0.865	1453.76	0.8287
DT-boosting	96.88	0.969	0.988	0.969	0.943	0.969	0.949	18.36	0.9341
SVM-boosting	79.17	0.846	0.784	0.792	0.776	0.772	0.7839	17938.87	0.5328
MLP-boosting	92.24	0.929	0.962	0.922	0.815	0.921	0.866	4374.49	0.832
LSTM	97.21	0.961	0.986	0.961	0.954	0.961	0.957	60794.46	0.9183

threshold of 0.5. The receiver operating characteristic curve with an AUC of 0.85 in Fig. 5. shows that the LSTM-Autoencoder model can perfectly differentiate between the two classes involved.

TABLE V
LSTM-AUTOENCODER RESULTS

Threshold	Accuracy	Precision	Recall	F1-measure
0.2	0.9015	0.899	0.8972	0.8991
0.3	0.9278	0.9211	0.9154	0.9285
0.4	0.9321	0.9374	0.9357	0.9318
0.5	0.9437	0.9451	0.9461	0.9472

The GA was used to investigate the best architectural factors, including the epoch, batch size, number of units, learning rate and dropout used as input to the LSTM network. The input data for this setup is the balanced Polish data set (smote application). The splitting ratio for the data set is 70% training data and 30% testing data. There are two models for the optimized LSTM.

1) LSTM-GA-PCA Model: The principal component analysis (PCA) was employed to obtain the top twenty-six features referred to as principal components (PCs) from the data set. Therefore, it reduced the number of features from 64 to 26 PCs. The total information embedded in the 26 PCs is 99.067; thus, the information loss due to PCA is 0.933%. The results of this model are given in Table VI and VII. We employed the optimal values in Table VI as the parameters for the optimised LSTM models. The ROC curve for the optimised LSTM-GA model is shown in Figure 6.

2) LSTM-GA Model: In the second model, the LSTM-GA model was applied to the 64 attributes of the data set. The results generated from the optimized LSTM are shown in Table VI. Similarly, the optimal architectural factors obtained are given in Table VII.

The results in Table VII show the effect of using a genetic algorithm to optimize an LSTM algorithm. The principal component analysis was employed to reduce the number of attributes. The results show that the LSTM-GA-PCA has the best performance in terms of accuracy and computation time. Hence the PCA and GA optimized the performance of the LSTM algorithm.

TABLE VI
OPTIMIZED LSTM ITERATION RESULTS

Epoch No	Batch Size	Neurons No	Learning rate	Dropout	Mean Square Error
310	20	120	0.1661	0.1783	0.0720
410	380	150	0.0003	0.2718	0.0591
10	120	20	0.1858	0.0065	0.0512
210	0	110	0.1814	0.1754	0.0495
210	0	60	0.0571	0.3986	0.0371
210	220	10	0.0066	0.3622	0.0999
310	360	180	0.1261	0.3962	0.09823

TABLE VII
OPTIMIZED LSTM OPTIMAL RESULTS

Setup	Epoch	Batch size	Neurons No	Learning-rate	Dropout	Loss	Accuracy
LSTM-GA	10	340	30	0.1069	0.3357	0.035926	98.03%
LSTM-GA -PCA	410	440	90	0.1275	0.0201	0.054666	98.11%

C. Adaptive Synthetic (ADASYN) and Upsample Minority Class

Two other data balancing techniques called adaptive synthetic (ADASYN), upsample minority class were applied to the data set. The proposed model LSTM-GA-PCA was employed for the classification of the data into bankrupt and non-bankrupt companies. Results show that the use of SMOTE methodology has improved the performance of the model considering the precision, recall and MCC. For instance, Fig 7-9 shows the recall, precision, and MCC results after the application of different balancing techniques. In addition, Fig. 10 depicts the ROC curves for the three data

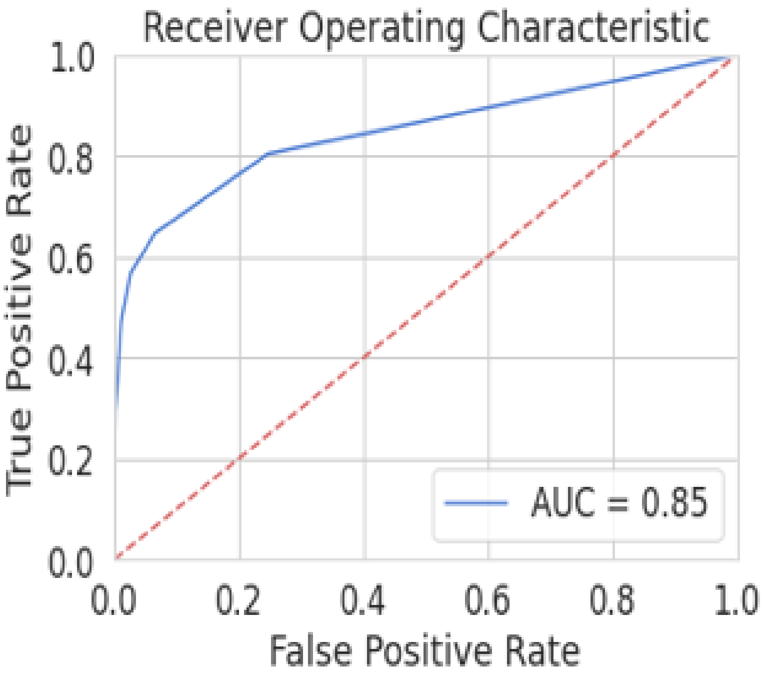

Fig. 5. LSTM-Autoencoder ROC Curve.

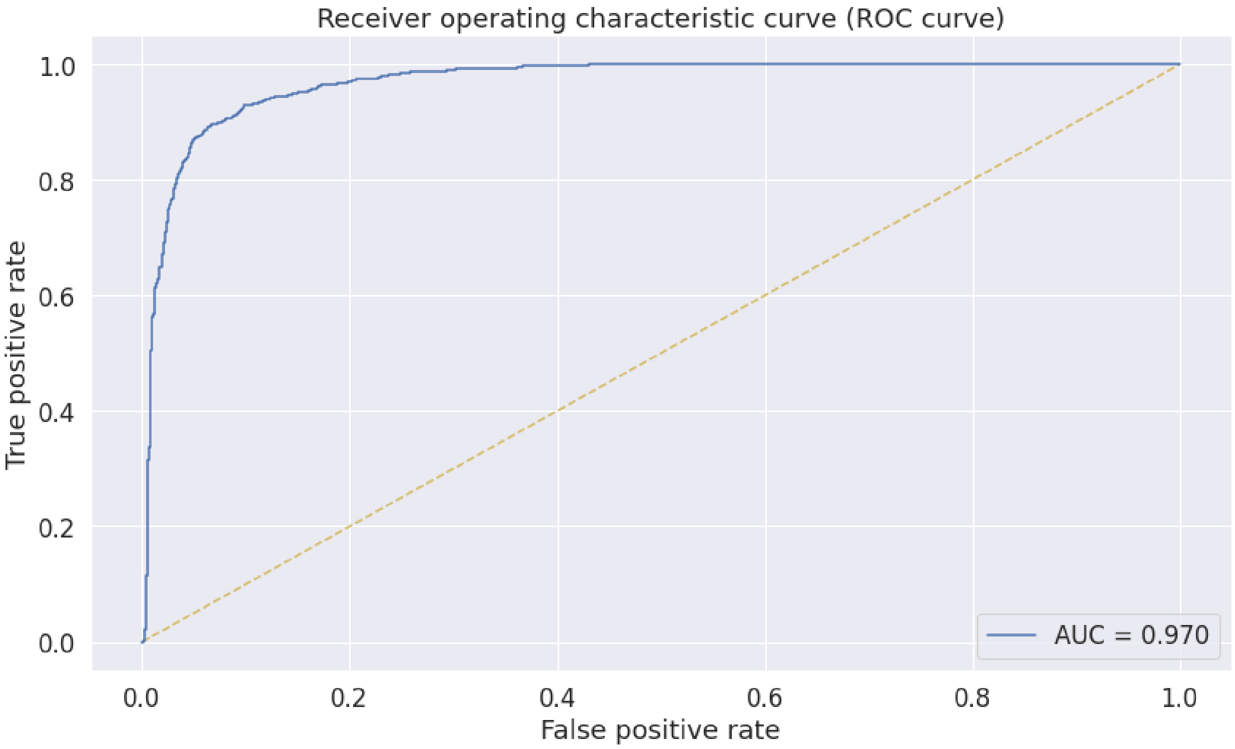

Fig. 6. LSTM-GA-PCA ROC Curve.

balancing techniques employed in this research work. Fig. 11-13 shows the confusion matrix of the improved model for different data balancing technique.

Figs. 2-4 show improved accuracy using different training data divisions (10-fold cross-validation, 90/10, 70/30, 50/50 training/testing ratios). They show that the 10-fold cross-validation has the best accuracy; boosting has the best accuracy for DT, while bagging performs better for MLP. On average, comparing the accuracy and AUC of all the models employed as shown in Table IV. The LSTM has the best performance with 97.21% followed by DT-boosting 97.01% and DT-bagging 96.45%. However, the two best-performing models in terms of AUC are DT-boosting and DT-bagging, with 0.990 and 0.988, respectively.

Fig. 5 and Table V show the results of the autoencoder model. Fig. 7 show the optimised LSTM. Comparative analysis of all the models from the experimental setup, the LSTM-autoencoder model has a good performance as the model threshold was increasing. The best performance from the LSTM-Autoencoder model has an accuracy of 94.37% compared to the proposed optimised LSTM-GA model with principal component analysis has the best performance in terms of 99.11% accuracy and loss of 0.054. Comparing the different data balancing techniques, Figs. 7-13 show an improvement in the modes. The best model was achieved through the application of SMOTE technique and the GA optimisation process.

V. Conclusions

In this paper, we have applied an oversampling technique called the synthetic minority oversampling technique (SMOTE) to solve the class imbalance problem of the Polish bankruptcy dataset. We applied the baseline models at different training and testing ratios. We further developed the ensemble classifiers from the baseline models. The performance of the experimental setup was evaluated using different performance metrics.

The research investigated single classifiers, and the results were compared to their ensemble models and the LSTM model. We applied different splitting ratios for the training and testing data set. An optimization algorithm was introduced to optimize and find the best hyperparameters for the LSTM model. The principal component analysis (PCA) was employed to resolve the high-dimensionality problem. An autoencoder algorithm was used to select the best feature before an LSTM model for the classification process.

The kappa statistic value of 0.918 for LSTM shows that the observations are in perfect agreement. The LSTM-GA model has the highest accuracy of 0.9811% and a low loss value of 0.054. A comparative analysis of the accuracy of the LSTM-GA and LSTM-Autoencoder models were carried out. We compared three different data balancing techniques to obtain the best model for the classification process. Finally, we concluded that the optimized model (LSTM-GA) with SMOTE technique gives better performance results than the ordinary LSTM model, autoencoder model and other models. All the experimental setups perform well in the range of 90% accuracy.

References

[1] A. Ansari, I. S. Ahmad, A. A. Bakar, and M. R. Yaakub, "A hybrid metaheuristic method in training artificial neural network for bankruptcy prediction," *IEEE Access*, vol. 8, pp. 176 640–176 650, 2020.

[2] U. Burkhanov, "The big failure: Lehman brothers' effects on global markets," *European Journal of Business and Economics*, vol. 2, pp. 17–20, 2011.

[3] Y. Cao, X. Liu, J. Zhai, and S. Hua, "A two-stage bayesian network model for corporate bankruptcy prediction," *International Journal of Finance & Economics*, vol. 27, no. 1, pp. 455–472, 2022.

[4] S. Ben Jabeur, N. Stef, and P. Carmona, "Bankruptcy prediction using the xgboost algorithm and variable importance feature engineering," *Computational Economics*, pp. 1–27, 2022.

[5] A. D. Aydin and S. C. Cavdar, "Prediction of financial crisis with artificial neural network: an empirical analysis on turkey," *International journal of financial research*, vol. 6, no. 4, pp. 36–45, 2015.

[6] C.-F. Tsai, "Financial decision support using neural networks and support vector machines," *Expert Systems*, vol. 25, no. 4, pp. 380–393, 2008.

[7] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications: A survey," *Applied Soft Computing*, vol. 93, p. 106384, 2020.

[8] C.-F. Tsai, Y.-F. Hsu, and D. C. Yen, "A comparative study of classifier ensembles for bankruptcy prediction," *Applied Soft Computing*, vol. 24, pp. 977–984, 2014.

[9] W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, "Machine learning in financial crisis prediction: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 421–436, 2011.

[10] H. Yan and H. Ouyang, "Financial time series prediction based on deep learning," *Wireless Personal Communications*, vol. 102, no. 2, pp. 683–700, 2018.

[11] G. Gilardoni, "Recurrent neural network models for financial distress prediction," *Master's thesis*, pp. 1–138, 2017.

[12] H. Chung and K.-s. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability*, vol. 10, no. 10, p. 3765, 2018.

[13] L. Bateni and F. Asghari, "Bankruptcy prediction using logit and genetic algorithm models: A comparative analysis," *Computational Economics*, vol. 55, no. 1, pp. 335–348, 2020.

[14] M. A. Albadr, S. Tiun, M. Ayob, and F. AL-Dhief, "Genetic algorithm based on natural selection theory for optimization problems," *Symmetry*, vol. 12, no. 11, p. 1758, 2020.

[15] H. Chung and K.-s. Shin, "Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction," *Neural Computing and Applications*, vol. 32, no. 12, pp. 7897–7914, 2020.

[16] S. C. Nayak, B. B. Misra, and H. S. Behera, "An adaptive second order neural network with genetic-algorithm-based training (asonn-ga) to forecast the closing prices of the stock market," *International Journal of Applied Metaheuristic Computing (IJAMC)*, vol. 7, no. 2, pp. 39–57, 2016.

[17] J. M. Bates and C. W. Granger, "The combination of forecasts," *Journal of the Operational Research Society*, vol. 20, no. 4, pp. 451–468, 1969.

[18] K. C. Lee, I. Han, and Y. Kwon, "Hybrid neural network models for bankruptcy predictions," *Decision Support Systems*, vol. 18, no. 1, pp. 63–72, 1996.

[19] S. Shetty, M. Musa, and X. Brédart, "Bankruptcy prediction using machine learning techniques," *Journal of Risk and Financial Management*, vol. 15, no. 1, p. 35, 2022.

[20] M. Brygała, "Consumer bankruptcy prediction using balanced and imbalanced data," *Risks*, vol. 10, no. 2, p. 24, 2022.

[21] M. E. Pérez-Pons, J. Parra-Dominguez, G. Hernández, E. Herrera-Viedma, and J. M. Corchado, "Evaluation metrics and dimensional reduction for binary classification algorithms: a case study on bankruptcy prediction," *The Knowledge Engineering Review*, vol. 37, p. e1, 2022.

[22] J. H. Min and C. Jeong, "A binary classification method for bankruptcy prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5256–5263, 2009.

[23] D. Veganzones and E. Séverin, "An investigation of bankruptcy prediction in imbalanced datasets," *Decision Support Systems*, vol. 112, pp. 111–124, 2018.

[24] S. Gite and K. Kotecha, "Evaluating the Impact of ANN Architecture for Driver Activity Anticipation in Semi-autonomous Vehicles." *Engineering Letters*, vol. 29, no. 3, pp. 873–880, 2021.

[25] S. Melina, H. Napitupulu, A. Sambas, A. Murniati, and V. A. Kusumaningtyas, "Artificial neural network-based machine learning approach to stock market prediction model on the indonesia stock

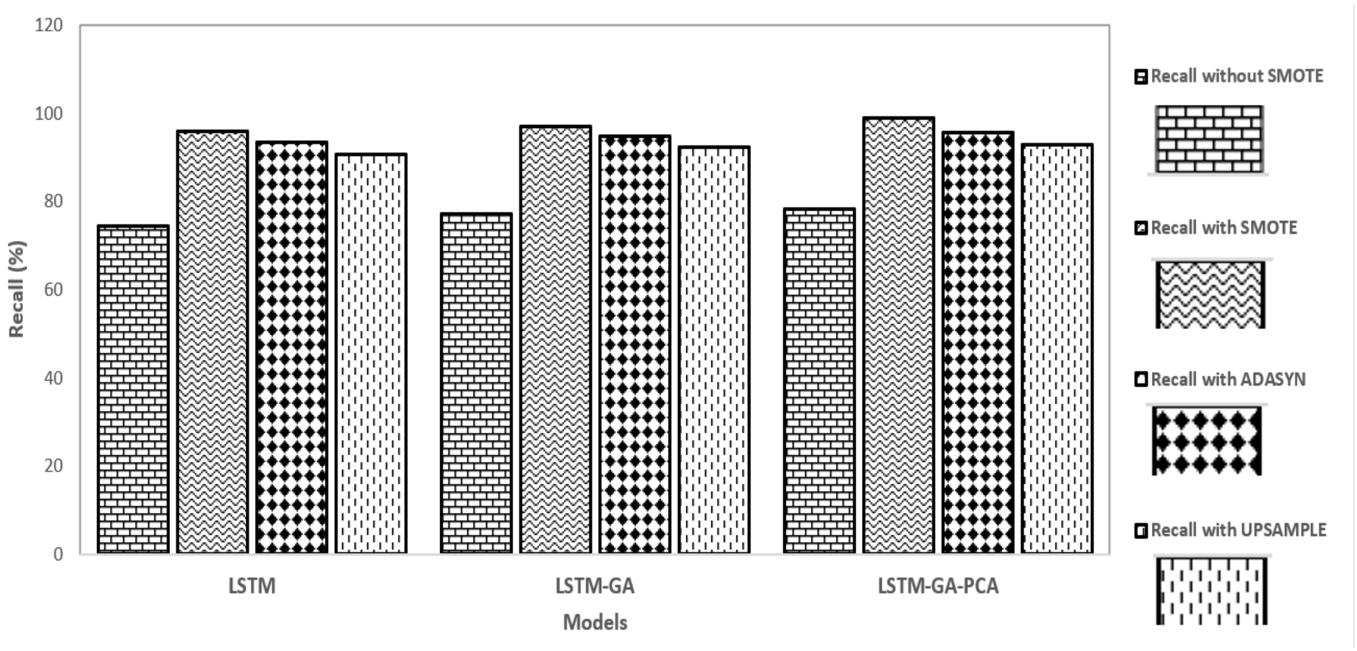

Fig. 7. Recall comparison

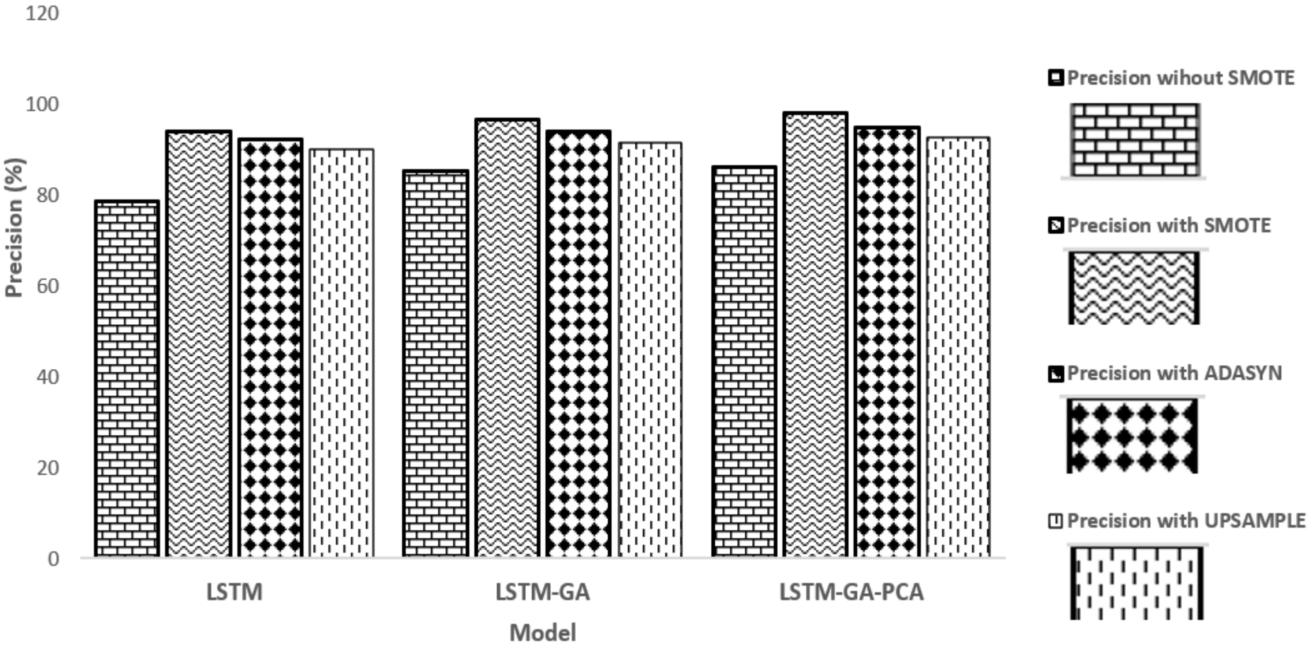

Fig. 8. Precision comparison

exchange during the covid-19," *Engineering Letters*, vol. 30, no. 3, pp. 988–1000, 2022.

[26] L. Hardinata, B. Warsito *et al.*, "Bankruptcy prediction based on financial ratios using jordan recurrent neural networks: a case study in polish companies," in *Journal of Physics: Conference Series*, vol. 1025, no. 1. IOP Publishing, 2018, p. 012098.

[27] L. Zhang, L. Luo, L. Hu, and M. Sun, "An svm-based classification model for migration prediction of beijing." *Engineering Letters*, vol. 28, no. 4, pp. 1023–1030, 2020.

[28] T.-N. Chou, "An explainable hybrid model for bankruptcy prediction based on the decision tree and deep neural network," in *2019 IEEE 2nd International Conference on Knowledge Innovation and Invention (ICKII)*. IEEE, 2019, pp. 122–125.

[29] N. Jiang, X. Zheng, H. Zheng, Q. Zheng *et al.*, "Long Short-Term Memory based PM2. 5 Concentration Prediction Method," *Engineering Letters*, vol. 29, no. 2, pp. 765–774, 2021.

[30] A. S. Girsang and D. Tanjung, "Fast Genetic Algorithm for Long Short-Term Memory Optimization." *Engineering Letters*, vol. 30, no. 2, pp. 528–536, 2022.

[31] W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, "Machine learning in financial crisis prediction: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 4, pp. 421–436, 2012.

[32] C. Liou, W. Cheng, C. Liou, and D. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, 2014.

[33] Z. Liu, H. Wang, W. Chen, L. Wang, and T. Li, "Bilateral discriminative autoencoder model orienting co-representation learning," *Knowledge-Based Systems*, vol. 245, p. 108653, 2022.

[34] N. T. N. Anh, T. Q. Khanh, N. Q. Dat, E. Amouroux, and V. K. Solanki, "Fraud detection via deep neural variational autoencoder oblique random forest," in *2020 IEEE-HYDCON*. IEEE, 2020, pp. 1–6.

[35] J.-R. Chang, L.-S. Chen, and L.-W. Lin, "A novel cluster based oversampling approach for classifying imbalanced sentiment data," *IAENG*

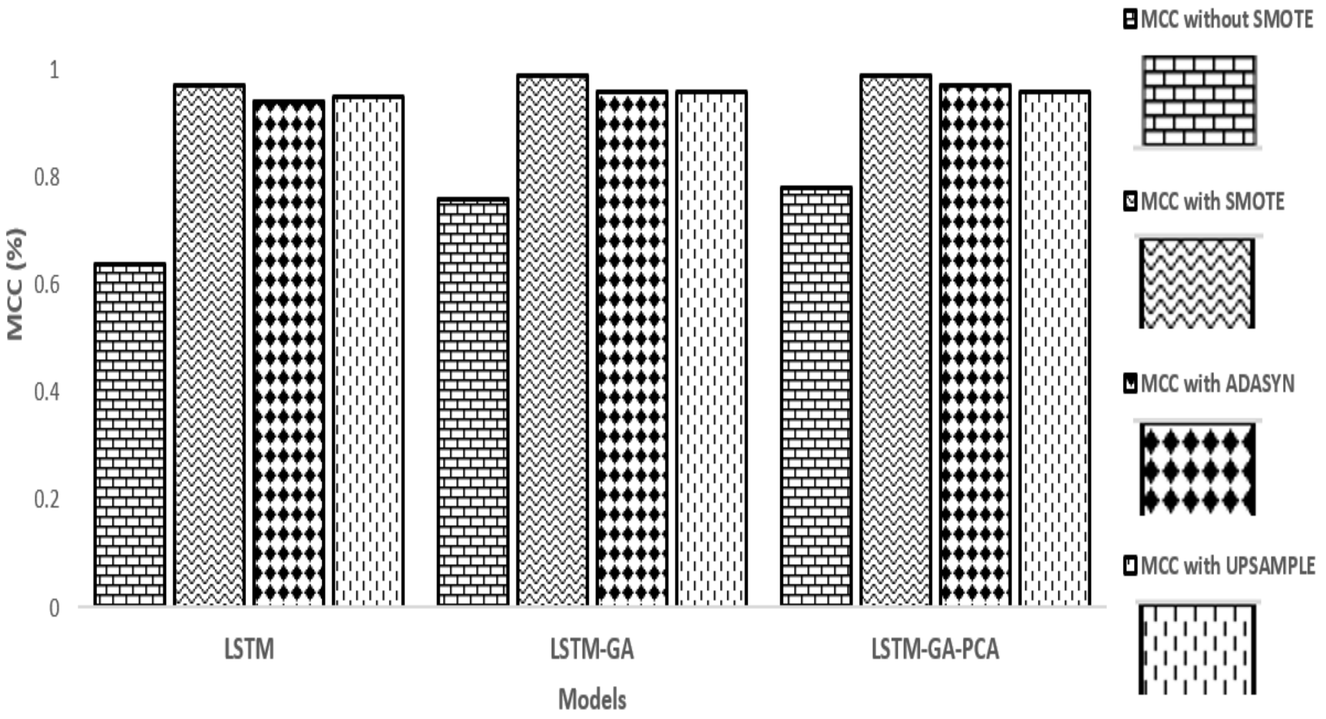

Fig. 9. MCC comparison

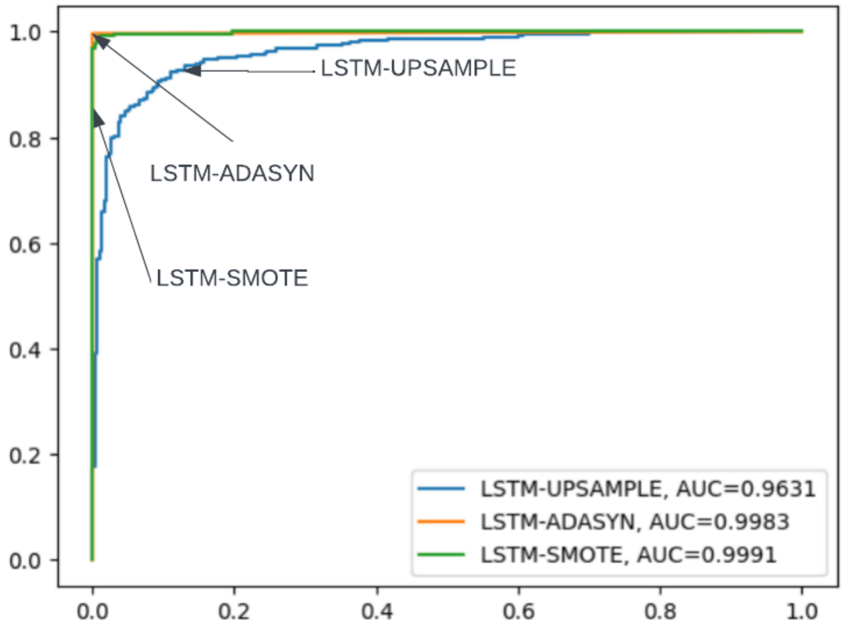

Fig. 10. ROC comparison

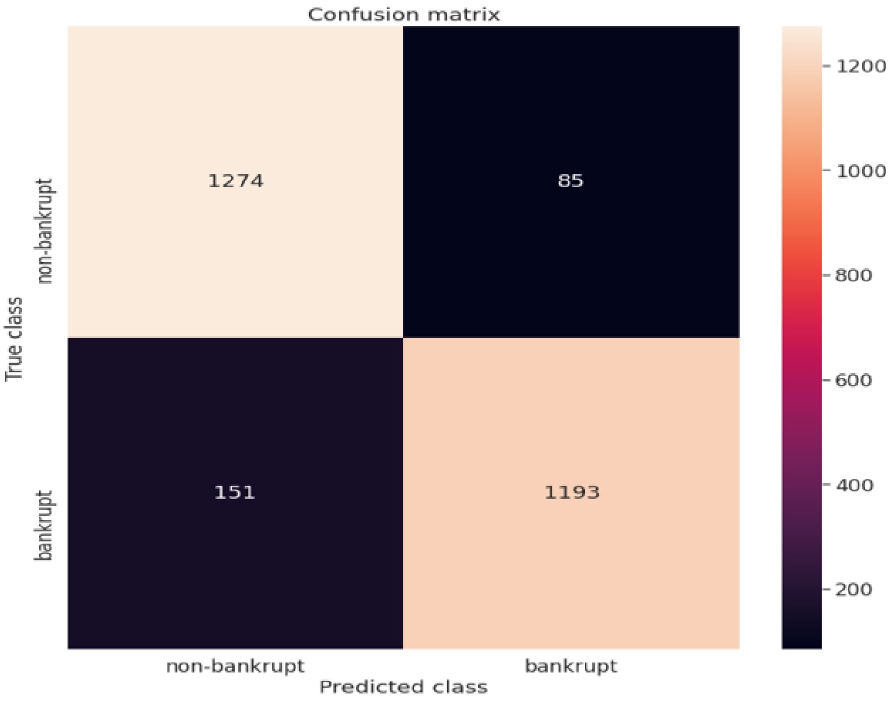

Fig. 11. LSTM-GA-PCA Confusion Matrix. It summarized the performance of the model. In general, the model miss-classified 145 features as false-positive and 95 features as false-negative. Hence, 91% of the data set was correctly classified.

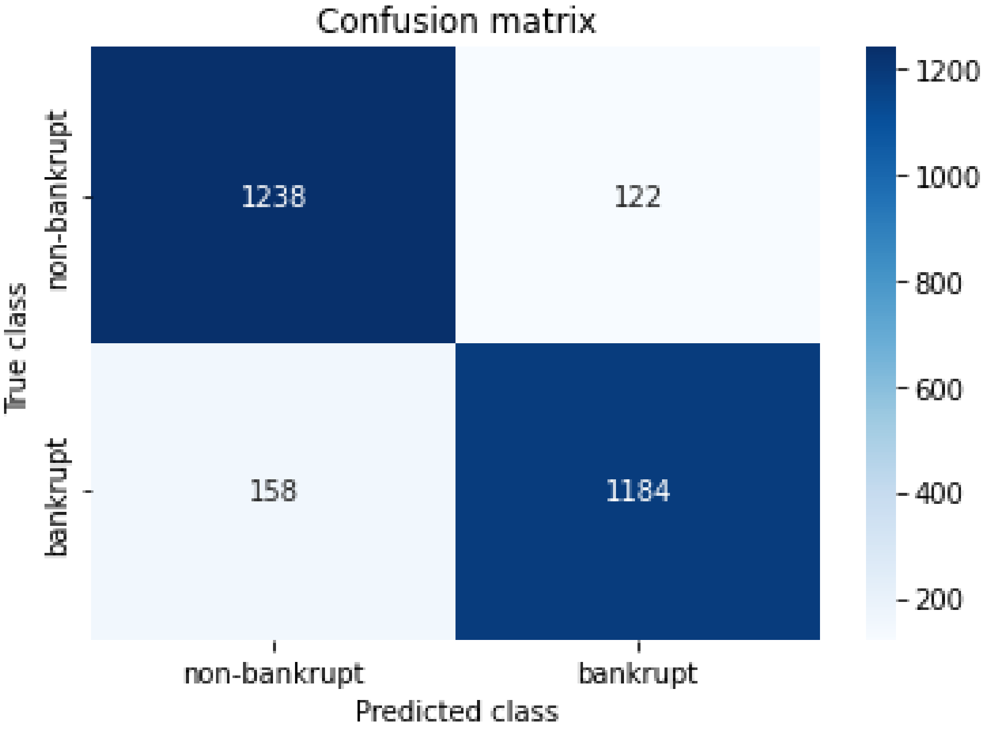

Fig. 12. LSTM-GA-PCA-ADASYN Confusion Matrix

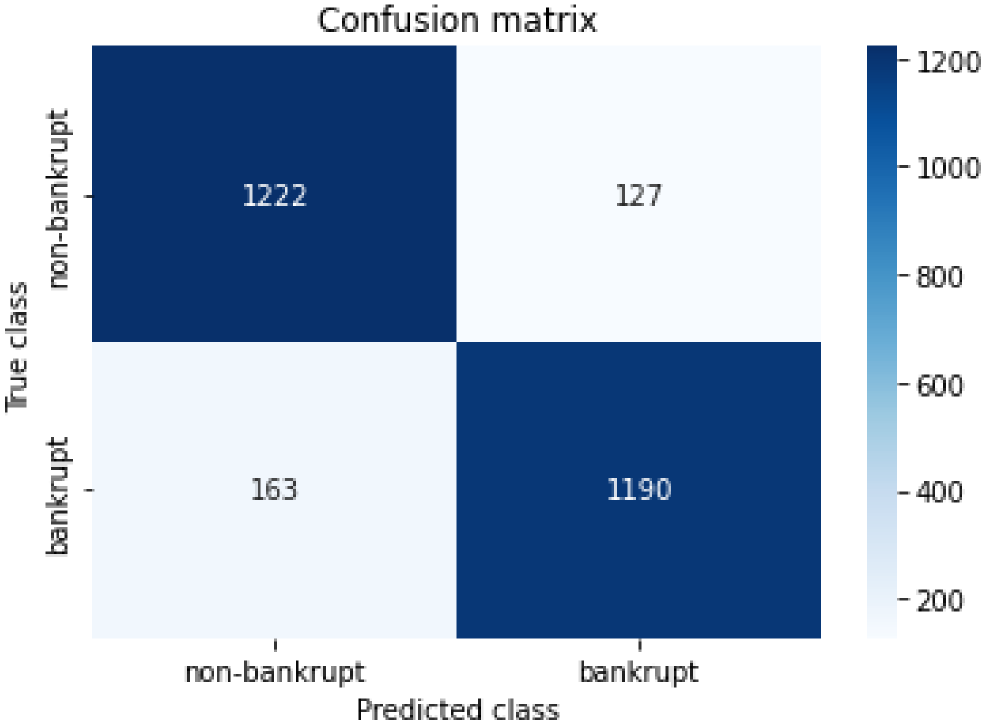

Fig. 13. LSTM-GA-PCA-UPSAMPLE Confusion Matrix

International Journal of Computer Science, vol. 48, no. 4, pp. 1118–1128, 2021.

[36] C. Bunkhumpornpat and S. Subpaiboonkit, “Safe level graph for synthetic minority over-sampling techniques,” in *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2013, pp. 570–575.

[37] L. Yu, L. Yu, and K. Yu, “A high-dimensionality-trait-driven learning paradigm for high dimensional credit classification,” *Financial Innovation*, vol. 7, no. 1, pp. 1–20, 2021.

[38] Z. Zhongwen and G. Huanghuang, “Visualization study of high-dimensional data classification based on pca-svm,” in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2017, pp. 346–349.

[39] D. Indriyanti and A. Dhini, “Clustering high-dimensional stock data using data mining approach,” in *2019 16th International Conference on Service Systems and Service Management (ICSSSM)*. IEEE, 2019, pp. 1–5.

[40] J. A. Adisa, S. O. Ojo, P. A. Owolawi, and A. B. Pretorius, “Financial distress prediction: Principle component analysis and artificial neural networks,” in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. IEEE, 2019, pp. 1–6.

[41] P. Lachheta and S. Bawa, “Combining synthetic minority oversampling technique and subset feature selection technique for class imbalance problem,” in *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, 2016, pp. 1–6.

[42] C. Zhang, Y. Zhou, J. Guo, G. Wang, and X. Wang, “Research on classification method of high-dimensional class-imbalanced datasets based on svm,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 7, pp. 1765–1778, 2019.

[43] A. Gosain and S. Sardana, “Handling class imbalance problem using oversampling techniques: A review,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017, pp. 79–85.

[44] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *biometrics*, pp. 159–174, 1977.

[45] E. Ileberi, Y. Sun, and Z. Wang, “Performance evaluation of machine learning methods for credit card fraud detection using smote and adaboost,” *IEEE Access*, vol. 9, pp. 165 286–165 294, 2021.