International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, Delaware, USA
Volume 13– No. 3, June 2026 – www.ijais.org

Loan Default Prediction System using Forward and Backward Propagation Techniques: A Comparative Study

Ismail Idowu Akuji
Department of Computer Science
Abdulrasaq Abubakar Toyin University, Oke-Ogba, Kwara State, Nigeria

Babajide Olanrewaju Ahmed
Department of Computer Science
University of Ibadan
Ibadan, Nigeria

Taofik Abiodun Ahmed
Department of Computer Science
Kwara State College of Arabic and Islamic Legal Studies
Ilorin, Nigeria

Idris Babatunde Adeyemi
Department of Computer Science
University of Ilorin
Ilorin, Nigeria

Ayodeji Jubril Alabi
Department of Computer Science
Kwara State University
Malete, Nigeria

ABSTRACT

In recent times, loan default has become a critical problem for financial institutions, underscoring the need for an accurate and reliable system that is capable of identifying vulnerable defaulters. This facilitates borrowers' creditworthiness assessment, which in turn assists in mitigating potential losses. Hence, this study proposes a comparative analysis of forward and backward propagation neural networks for predicting loan defaulters. The dataset used for the models was sourced from the Kaggle repository, and it consists of 255,347 instances and 17 features, which were undersampled to 47,444, 50% Class 0 and 50% Class 1, due to the class distribution being imbalanced. The analysis of variance (ANOVA) was employed to identify the set of features that may or may not influence the target feature. K-fold cross-validation was applied to assess the robustness and generalization ability of the proposed models. Results show that both models improved substantially after cross-validation, especially in terms of accuracy and loss. The FPNN increased from 0.7110 baseline accuracy to 0.8861 mean cross-validated accuracy. At the same time, the BPNN improved from 0.6920 to 0.8858, indicating that cross-validation produced a more stable and reliable estimate of model performance. Similarly, the loss values dropped from 0.584308 to 0.314693 for FPNN and from 0.605187 to 0.314709 for BPNN, which suggests better learning and stronger generalization after validation across multiple folds. The findings of this study highlight the potential of machine learning techniques in improving loan default prediction while reducing lending risk, particularly the efficacy of cross-validation in affirming the robustness and generalization of machine learning models. Future studies can build on this research by using different datasets and integrating hyperparameter tuning to further improve models, especially in terms of precision, thereby contributing to an effective, reliable, and deployable loan default prediction model.

Keywords

Loan default, risk assessment, ANOVA, forward propagation neural network, backward propagation neural network, cross-validation.

1. INTRODUCTION

The widespread demand for loans is a pressing concern in society, particularly in Nigeria. The economic downfall poses serious adversities, making small and medium-sized enterprises and individuals heavily rely on borrowing and lending from financial institutions. According to [1], loan default has remained a dominant issue, resulting from a poor loan management system. However, several studies have been conducted, and systems have been developed to mitigate against loan default. Machine learning has shown significant promise to intervene to loan default prevalence in society. For instance, [2], [3], [4], and [5] investigated the application of machine learning for predicting loan defaulters; however, their studies have demonstrated the efficacy of the machine learning model.

Despite the effort made by the previous studies, gaps are left to be addressed. Firstly, the issue of overfitting in existing studies poses significant issues in prediction results, which often leads to significant financial losses for lenders. In addition, existing studies failed to exhibit the key factors that influence loan default, especially. This study, however, compares forward and backward propagation neural network techniques for the development of the proposed models while employing ANOVA for the identification of the important features in identifying loan defaulters. The proposed models not only enhance prediction accuracy but also provide a more robust and reliable solution for financial institutions to tackle loan default prevalence. This study aims to answer the following questions:

1. What are the key factors that influence loan default?

2. What are the effects of the identified key factors on loan default?

3. How effective are the proposed models in predicting vulnerable defaulters?

2. LITERATURE REVIEW

Today, the identification of loan defaulters has been a vital area of study. Recently, machine learning techniques have been extensively explored by various studies to mitigate against loan default prevalence. A study by [6] evaluates the performance of various machine learning algorithms for predicting credit default.

The study employed a Spanish bank dataset and various algorithms, including XGBoost. The authors found that XGBoost outperformed other models. Similarly, [7] explored the application of machine learning algorithms for microcredit scoring in markets where credit history is limited and found that AdaBoost, XGBoost, and Random Forest classifiers achieved high prediction accuracy (81%). Reference [8] developed a predictive method for identifying default customers in the Egyptian banking industry using machine learning and found that feature selection significantly improved accuracy, with the Decision Tree achieving the highest accuracy of 94.85%. Additionally, [9] compared the performance of decision tree and random forest algorithms in predicting loan defaults. The results showed that the Random Forest classifier outperformed the Decision Tree classifier, achieving an accuracy score of 80% compared to 73%. [2] employed machine learning models to predict loan behavior and prevent defaults, achieving an accuracy of 86.17% using Extra Trees Classifier.

Furthermore, [10] proposed a loan default prediction model based on knowledge graph technology. The model achieves 96.79% accuracy using XGBoost + knowledge graph. However, the application of neural networks, such as forward and backward propagation techniques, remains underexplored among recent studies. Although, [11] proposed a back propagation neural network for predicting online loan users, achieving 98.01% accuracy in predicting loan defaults. Additionally, [12] also investigated the efficacy of Genetic-based back propagation neural network for credit risk assessment. The finding underscores that the GA-BPNN efficacy in effectively classifying the underlying default risk and credit ratings based on an average accuracy of 89%. Reference [13] also studied a credit risk prediction model hybrid of backpropagation and random forest. The model achieved an accuracy score of 95.1%, proving the robustness of the model for credit risk prediction.

While the state of the art in loan default prediction has evolved significantly in recent years, with the development of more sophisticated models based on neural networks. Notably, this study enhances the study of [11] by proposing a comparative approach to loan default prediction using ANOVA and forward and backward propagation techniques. The proposed model aims to improve prediction accuracy and provide a more robust and reliable solution for financial institutions.

3. METHODOLOGY

3.1 Framework of the proposed models

The framework of the proposed model is depicted in Figure 1. The process begins with data input, followed by data preprocessing, model development, and evaluation.

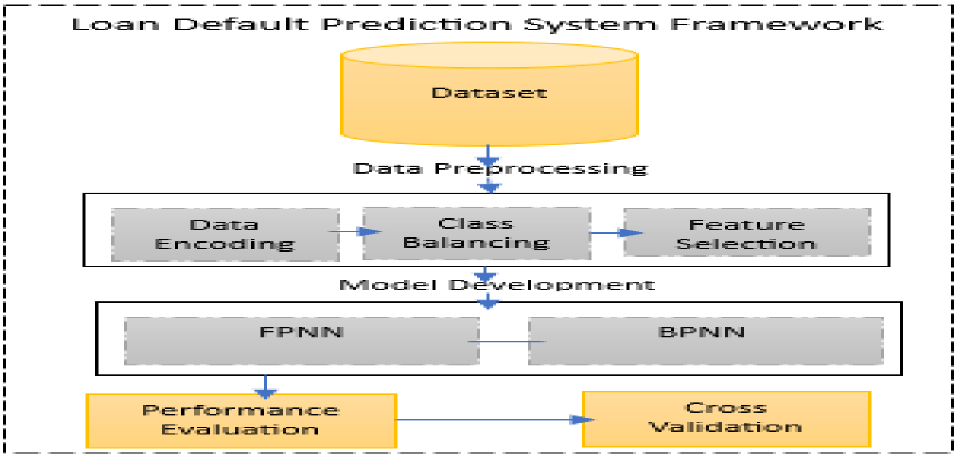
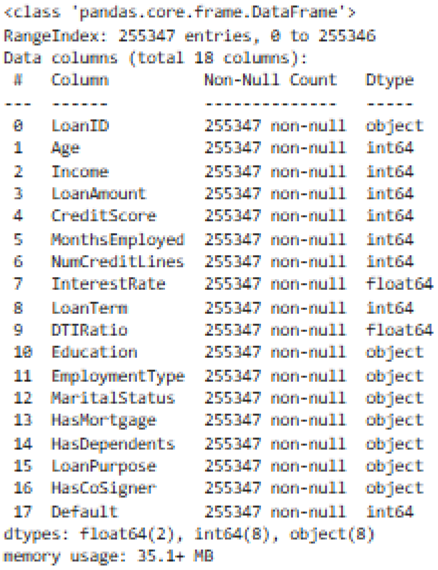


Fig 1: Framework of the proposed loan prediction system

3.2 Dataset Description and Preprocessing

A dataset consisting of 255347 instances and 17 features, obtained from the Kaggle repository, is proposed for this study. The description of the dataset, shown in Figure 2, reveals that the dataset contains no missing values. However, the dataset includes categorical variables that need to be preprocessed to make them readily available for model training. This is because machine learning models typically require numerical datasets.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255347 entries, 0 to 255346
Data columns (total 18 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   LoanID          255347 non-null  object 
 1   Age             255347 non-null  int64  
 2   Income          255347 non-null  int64  
 3   LoanAmount      255347 non-null  int64  
 4   CreditScore     255347 non-null  int64  
 5   MonthsEmployed  255347 non-null  int64  
 6   NumCreditLines  255347 non-null  int64  
 7   InterestRate    255347 non-null  float64
 8   LoanTerm        255347 non-null  int64  
 9   DTIRatio        255347 non-null  float64
 10  Education       255347 non-null  object 
 11  EmploymentType  255347 non-null  object 
 12  MaritalStatus   255347 non-null  object 
 13  HasMortgage     255347 non-null  object 
 14  HasDependents   255347 non-null  object 
 15  LoanPurpose     255347 non-null  object 
 16  HasCoSigner     255347 non-null  object 
 17  Default         255347 non-null  int64  
dtypes: float64(2), int64(8), object(8)
memory usage: 35.1+ MB
```

Fig 2: Dataset description

The preprocessing mechanisms are described as follows:

- Data encoding:

Features such as education, employment type, marital status, mortgage, dependency, loan purpose, and cosigner are encoded to numerical values.

- Class balancing

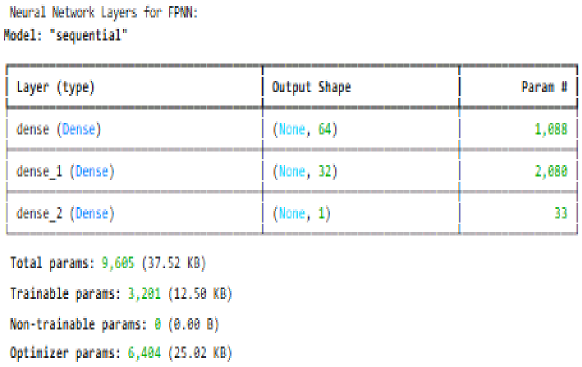
The dataset is imbalanced, so Random Undersampling was used to downsample the majority class to the size of the minority class.

- Feature selection:

In this study, analysis of variance (ANOVA) is proposed to test interactions between the dataset's features to determine which feature may or may not influence the target variable.

3.3 Model Development

The neural network model for FPNN, as shown in Figure 3, is a sequential model consisting of three dense layers. The first layer has 64 neurons and takes the input features, the second layer has 32 neurons, and the third layer has a single output. The model has a total of 9,605 parameters, with 3,201 being trainable and the rest being optimizer parameters. This architecture is designed for binary classification, given the single output neuron with a presumed sigmoid activation function.



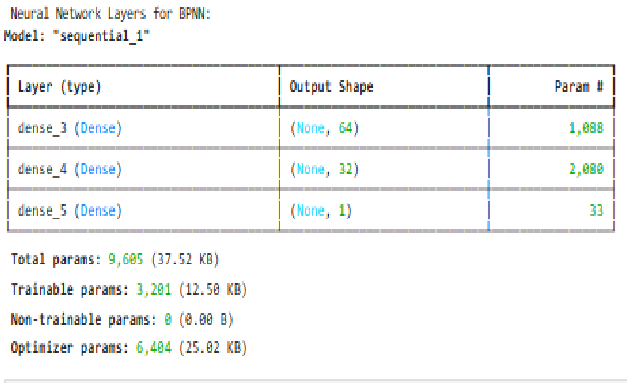
```
Neural Network Layers for FPNN:
Model: "sequential"

| Layer (type)       | Output Shape  |   Param # |
|--------------------|---------------|-----------|
| dense (Dense)      | (None, 64)    |     1,088 |
| dense_1 (Dense)    | (None, 32)    |     2,080 |
| dense_2 (Dense)    | (None, 1)     |        33 |

Total params: 9,605 (37.52 KB)
Trainable params: 3,201 (12.50 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 6,404 (25.02 KB)
```

Fig 3: Forward propagation neural network layers

Furthermore, the neural network model for BPNN, shown in Figure 4, consists of three dense layers, with an architecture similar to the FPNN model. It consists of a 64-neuron layer for feature extraction, a 32-neuron layer for further processing, and a single-neuron output layer. The model has a total of 9,605 parameters, with 3,201 trainable parameters and 6,404 optimizer parameters, indicating a binary classification setup with a likely sigmoid output activation.



```
Neural Network Layers for BPNN:
Model: "sequential_1"

| Layer (type)       | Output Shape  |   Param # |
|--------------------|---------------|-----------|
| dense_3 (Dense)    | (None, 64)    |     1,088 |
| dense_4 (Dense)    | (None, 32)    |     2,080 |
| dense_5 (Dense)    | (None, 1)     |        33 |

Total params: 9,605 (37.52 KB)
Trainable params: 3,201 (12.50 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 6,404 (25.02 KB)
```

Fig 4: Backward propagation neural network layers

3.4 Performance Evaluation Techniques

Accuracy and loss metrics are proposed to measure the proposed models. Accuracy measures the frequency of correct predictions, while Loss quantifies the error margin between the predicted and actual target values, with lower loss indicating better model fit. Additionally, a confusion matrix was used to provide a detailed breakdown of model performance by showing True Positives, True Negatives, False Positives, and False Negatives. From the confusion matrix, derived metrics such as Precision, Recall, and F1-Score were computed to evaluate the model's ability to handle class imbalance and minimize both false alarms and missed detections, with the formulae given as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \ldots\ldots\ldots\ldots\ldots (Equation\ 1)$$

$$Precision = \frac{TP}{TP + FP} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (Equation\ 2)$$

$$Recall = \frac{TP}{TP + FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (Equation\ 3)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \ldots\ldots (Equation\ 4)$$

Where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively

Finally, the AUC-ROC curve was used to assess the model's discriminative ability across all classification thresholds. AUC summarizes the trade-off between True Positive Rate and False Positive Rate, where a score closer to 1 indicates better separation between classes.

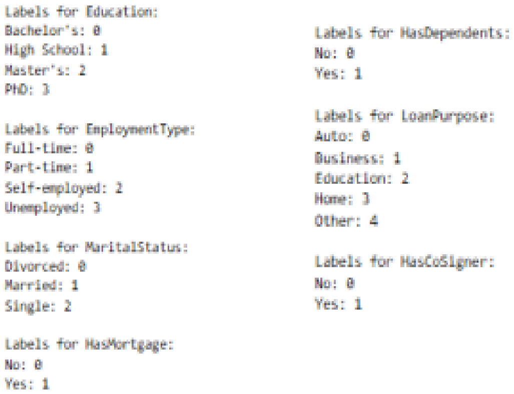
3.5 Cross-Validation Process

Cross-validation was used to assess the robustness and generalization ability of the FPNN and BPNN models. A stratified 5-fold scheme was applied so that each fold preserved the original class distribution, which is especially important for imbalanced classification problems. In each iteration, the model was trained on four folds and validated on the remaining fold, and this process was repeated until every subset had served once as validation data. The mean accuracy and mean loss across all folds were then computed to provide a more reliable performance estimate than a single train-test split. This approach also helped reduce the risk of performance being influenced by a favorable or unfavorable data split. Early stopping was incorporated during training to prevent overfitting by monitoring validation loss and restoring the best weights when improvement stalled.

4. RESULTS AND DISCUSSION

4.1 Results from Data Exploratory Analysis

The result of data encoding is shown in Figure 5, which illustrates the categorical values encoded to numerical values.



```
Labels for Education:
Bachelor's: 0
High School: 1
Master's: 2
PHD: 3

Labels for EmploymentType:
Full-time: 0
Part-time: 1
Self-employed: 2
Unemployed: 3

Labels for MaritalStatus:
Divorced: 0
Married: 1
Single: 2

Labels for HasMortgage:
No: 0
Yes: 1

Labels for HasDependents:
No: 0
Yes: 1

Labels for LoanPurpose:
Auto: 0
Business: 1
Education: 2
Home: 3
Other: 4

Labels for HasCoSigner:
No: 0
Yes: 1
```

Fig 5: Result of data encoding

4.2 Results from Class Balancing

The class distribution in Figure 6 indicates a clear imbalance in the target variable, with class 0 (non-default) accounting for 225,694 records or 88.39% of the dataset, while class 1 (default) has only 29,653 records, representing 11.61%. This suggests that the dataset is heavily dominated by class 0, which may cause a predictive model to favor the majority class and perform poorly on the minority class if the imbalance is not addressed.

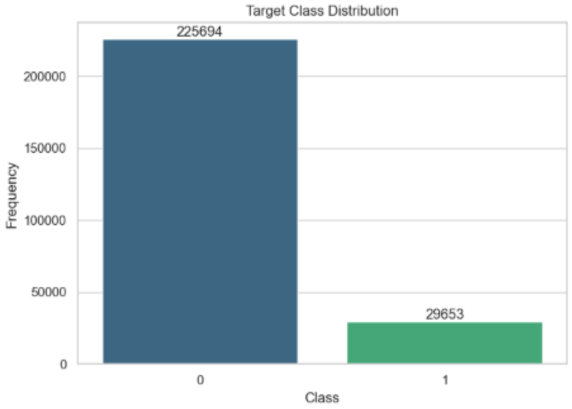


Fig 6: Raw data (target) class distribution

Moreover, the dataset after splitting shows that, before undersampling, the training set was also highly imbalanced, with class 0 containing 180,555 samples and class 1 containing only 23,722 samples. After applying random undersampling, both classes were reduced to an equal size of 23,722 samples each, resulting in a balanced training set (Figure 7).

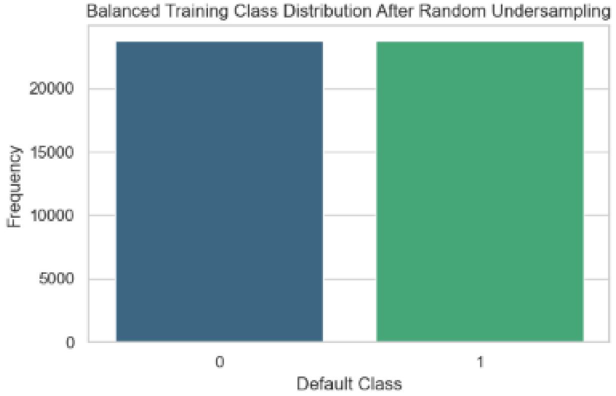


Fig 7: Balanced training class (target) distribution after random undersampling

This balance helps the model learn both classes more fairly and improves its ability to detect the minority class during training.

4.3 Results from Statistical Analysis

Some selected features were statistically analyzed to provide deeper insight into the dataset (see Table 1). The features examined include income, loan amount, credit score, and interest rate.

Table 1. Statistical analysis of some selected features

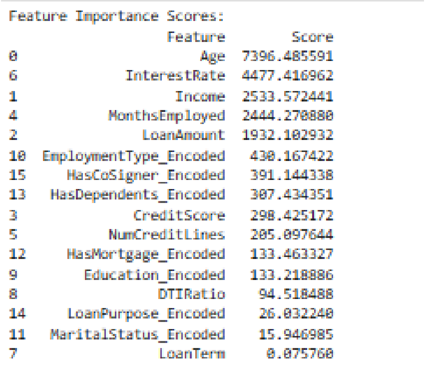
Feature	Count	Mean	Std	Min	Max	Variance	Skewness	Kurtosis
Income	255347.0	82499.30	38963.01	15000.0	149999.0	1.518116e+09	-0.000381	-1.198361
LoanAmount	255347.0	127578.87	70840.71	5000.0	249999.0	5.018406e+09	-0.001827	-1.203680
CreditScore	255347.0	574.26	158.90	300.0	849.0	2.525044e+04	0.004688	-1.200302
InterestRate	255347.0	13.49	6.64	2.0	25.0	4.404238e+01	0.004608	-1.197167

The statistical analysis in Table 1 shows that the selected features are fairly well spread across the dataset with 255,347 observations for each variable, indicating no missing-count issue in these columns. Income has a mean of 82,499.30 and a median of 82,466, while LoanAmount has a mean of 127,578.87 and a median of 127,556, suggesting that both variables are centered close to their middle values and are not strongly skewed. CreditScore has an average of 574.26 with values ranging from 300 to 849, showing a wide spread in borrower credit quality. InterestRate has a mean of 13.49 and a median of 13.46, indicating that most loans cluster around moderate interest levels, with some higher-rate loans extending up to 25.0. The standard deviations for Income and LoanAmount are also relatively large, which reflects substantial variation among applicants.

The skewness values for all four features are very close to zero, which implies that their distributions are approximately symmetric. In contrast, the kurtosis values are around -1.20 for each feature, suggesting flatter-than-normal distributions with lighter tails and fewer extreme outliers than a normal distribution would typically show. Overall, these statistics indicate that the selected features are numerically stable, broadly distributed, and suitable for modeling after proper scaling.

4.4 Results from Feature Selection Process

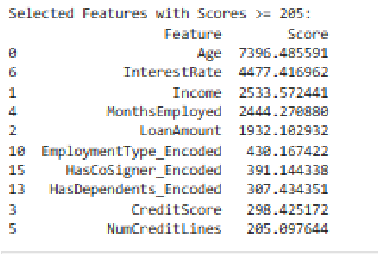
Furthermore, Figure 8 displays the feature importance scores for all the features present in the dataset used.



```
Feature Importance Scores:
                     Feature        Score
0                        Age  7396.485591
6               InterestRate  4477.416962
1                     Income  2533.572441
4             MonthsEmployed  2444.270880
2                 LoanAmount  1932.102932
10  EmploymentType_Encoded     430.167422
15     HasCoSigner_Encoded     391.144338
13   HasDependents_Encoded     307.434351
3                CreditScore   298.425172
5             NumCreditLines   205.097644
12      HasMortgage_Encoded    133.463327
9         Education_Encoded    133.218886
8                   DTIRatio    94.518488
14     LoanPurpose_Encoded      26.032240
11   MaritalStatus_Encoded      15.946985
7                   LoanTerm     0.075760
```

Fig 8: Feature importance scores

A threshold of 200 is set to select the important features. The 10 topmost features are depicted in Figure 9.



```
Selected Features with Scores >= 205:
                     Feature        Score
0                        Age  7396.485591
6               InterestRate  4477.416962
1                     Income  2533.572441
4             MonthsEmployed  2444.270880
2                 LoanAmount  1932.102932
10  EmploymentType_Encoded     430.167422
15     HasCoSigner_Encoded     391.144338
13   HasDependents_Encoded     307.434351
3                CreditScore   298.425172
5             NumCreditLines   205.097644
```

Fig 9: Selected 10 topmost features

4.5 Model Performance Evaluation Results

This subsection presents and discusses the performance evaluation results of each model using the metrics defined previously. Figure 10 presents the training and validation accuracy of the FPNN model using the selected feature set. The training accuracy increased slightly from approximately 0.674 at iteration 0 to about 0.683 at iteration 4, indicating a gradual improvement during training. The validation accuracy showed greater variation, increasing from approximately 0.681 to a peak of 0.711 at iteration 1, before decreasing to about 0.677 at iteration 3 and rising slightly to 0.685 at iteration 4. Overall, the results suggest that the model achieved modest training improvement, while the validation accuracy fluctuated across iterations.

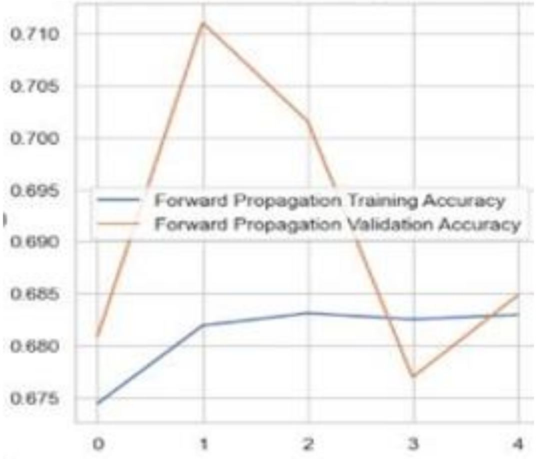


Fig 10: Forward propagation neural network accuracy

Figure 11 presents the training and validation loss of the FPNN model using the selected feature set. The training loss decreased gradually from approximately 0.601 at iteration 0 to 0.591 at iteration 4, indicating steady improvement during training. In contrast, the validation loss fluctuated across iterations, decreasing from about 0.585 at iteration 0 to its lowest value of approximately 0.557 at iteration 1, before increasing sharply to around 0.606 at iteration 3 and then declining slightly at iteration 4. Overall, the results indicate that while the model showed a consistent reduction in training loss, the validation loss varied noticeably, suggesting less stable generalization performance.

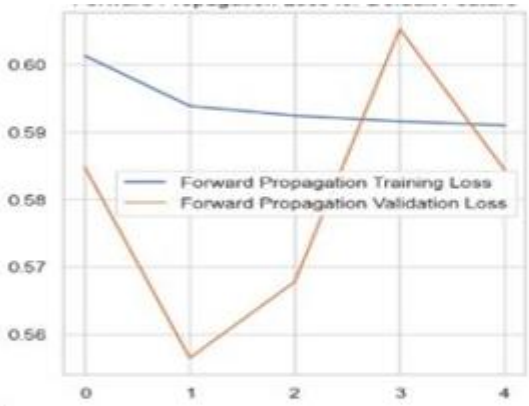


Fig 11: Forward propagation neural network loss

Figure 12 shows the confusion matrix for the FPNN model. The model correctly classified 32,449 samples as class 0 and 3,863 samples as class 1. However, 12,690 class 0 samples were incorrectly predicted as class 1, while 2,068 class 1 samples were incorrectly predicted as class 0. This indicates that the model performs better at identifying class 0 than class 1, but it still makes a noticeable number of misclassifications.

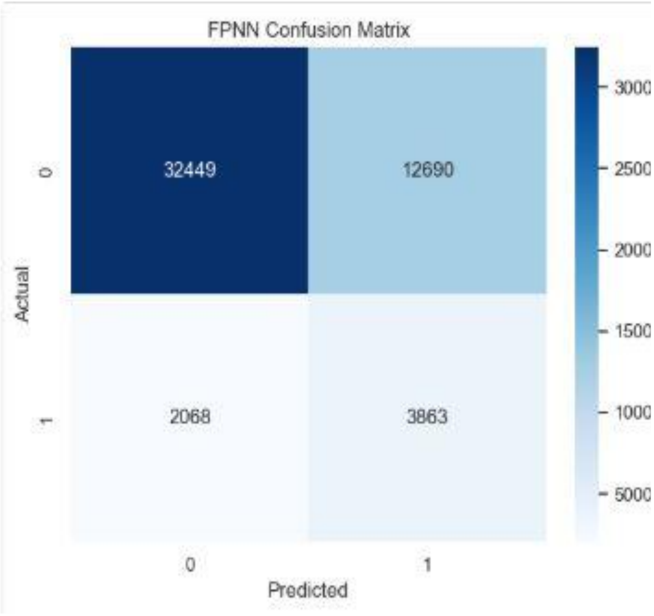


Fig 12: Forward propagation neural network confusion matrix

From the confusion matrix in Figure 12, the following data are obtained:

- True Negatives (TN): 32,449 (class 0 correctly predicted as class 0)
- True Positives (TP): 3,863 (class 1 correctly predicted as class 1)
- False Positives (FP): 12,690 (class 0 incorrectly predicted as class 1)
- False Negatives (FN): 2,068 (class 1 incorrectly predicted as class 0)

Using the formulae provided in Section 3 of this paper, the results are presented in Table 2.

Table 2. FPNN performance evaluation results

Metric	Value	Percentage
Accuracy	0.7110	71.10%
Precision	0.2334	23.34%
Recall	0.6513	65.13%
F1-Score	0.3420	34.20%

The FPNN model, as shown in Table 2, achieves a moderate accuracy of 71.10% while suffering from critically low precision (23.34%) despite reasonable recall (65.13%). This means the model captures about two-thirds of actual class 1 samples, but generates excessive false positives when it predicts class 1; it is correct only about 1 in 4 times. The low F1-score (34.20%) confirms this severe precision-recall imbalance and demonstrates that accuracy is a misleading metric for this model. The model appears biased toward predicting the minority class.

Figure 13 presents the training and validation accuracy of the BPNN model using the selected feature set. The training accuracy increased gradually from approximately 0.673 at iteration 0 to 0.686 at iteration 6, indicating steady improvement during training. The validation accuracy showed more variation, rising from approximately 0.667 at iteration 0 to a peak of about 0.690 at iteration 3, before decreasing to around 0.672 at iteration 5 and slightly increasing at iteration 6. Overall, the results indicate that the model achieved a consistent improvement in training accuracy, while the validation accuracy fluctuated across iterations, suggesting less stable generalization performance.

Fig 13: Backward propagation neural network accuracy

Figure 14 presents the training and validation loss of the BPNN model using the selected feature set. The training loss decreased gradually from approximately 0.602 at iteration 0 to 0.589 at iteration 6, indicating consistent improvement during training. In contrast, the validation loss showed noticeable fluctuations, decreasing from approximately 0.618 at iteration 0 to about 0.592 at iteration 1, then increasing at iteration 2 before reaching its lowest value of approximately 0.580 at iteration 3. Afterward, the validation loss increased again and ended at around 0.605 at iteration 6. Overall, the results indicate that the model achieved a steady reduction in training loss, while the validation loss varied across iterations, suggesting less stable generalization performance.

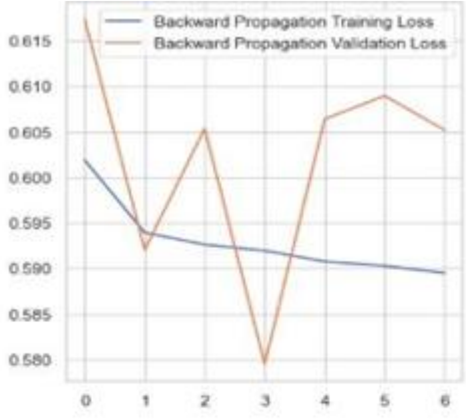


Fig 14: Backward propagation neural network loss

Figure 15 shows the confusion matrix for the BPNN model. The model correctly classified 31,305 samples as class 0 and 4,032 samples as class 1. It incorrectly predicted 13,834 class 0 samples as class 1, while 1,899 class 1 samples were predicted as class 0. This indicates that the BPNN model identifies class 1 slightly better than the FPNN model, but it has more incorrect predictions for class 0.

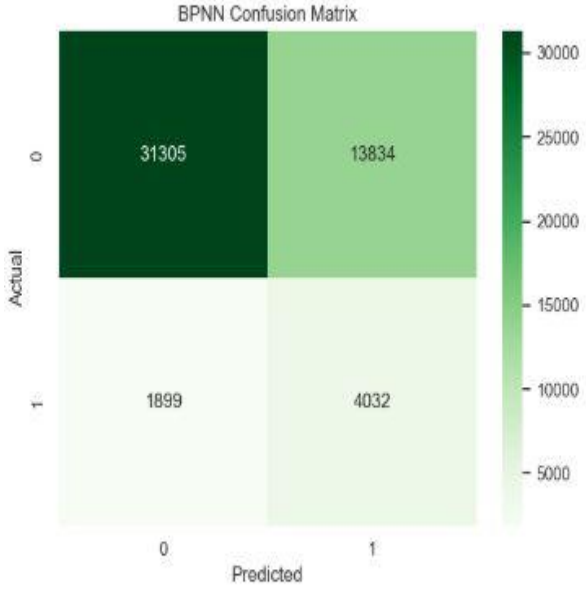


Fig 15: Backward propagation neural network confusion matrix

From the confusion matrix in Figure 15, the following data are obtained:

- True Negatives (TN): 31,305 (class 0 correctly predicted as class 0)
- True Positives (TP): 4,032 (class 1 correctly predicted as class 1)
- False Positives (FP): 13,834 (class 0 incorrectly predicted as class 1)
- False Negatives (FN): 1,899 (class 1 incorrectly predicted as class 0)

Using the formulae provided in Section 3 of this paper, the results are presented in Table 3.

Table 3. BPNN performance evaluation results

Metric	Value	Percentage
Accuracy	0.6920	69.20%
Precision	0.2257	22.57%
Recall	0.6800	68.00%
F1-Score	0.3390	33.90%

The BPNN model in Table 3 achieves a moderate accuracy of 69.20% while suffering from critically low precision (22.57%) despite reasonable recall (68.00%).

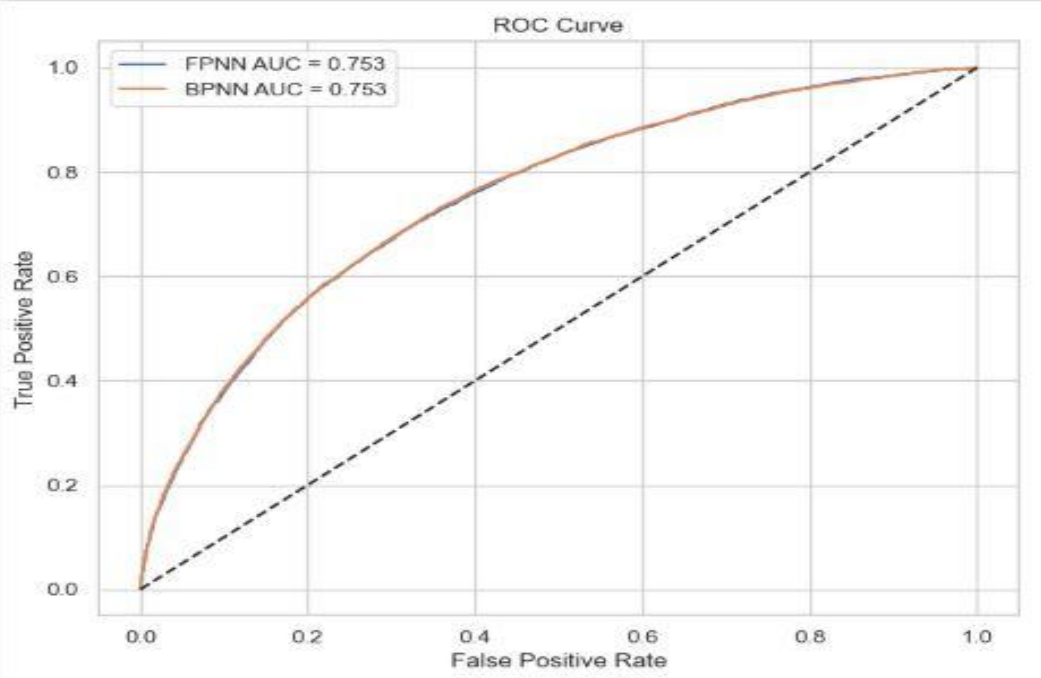


Fig 16: ROC curve for forward-backward propagation neural network models

4.6 Results from Cross-Validation (CV) of Models

Figure 17 shows the cross-validated confusion matrix for the FPNN model. The model correctly classified 224,472 samples as class 0 and 1,794 samples as class 1. It incorrectly classified 1,222 class 0 samples as class 1, while 27,859 class 1 samples were predicted as class 0. This shows that the cross-validated FPNN model performs very well in identifying class 0, but it struggles to correctly classify class 1 samples.

This means the model captures about 68% of actual class 1 samples, but generates excessive false positives when it predicts class 1; it is correct only about 1 in 4 times. The low F1-score (33.90%) confirms this severe precision-recall imbalance and is even slightly worse than the FPNN's F1-score (34.20%), indicating marginally inferior overall performance. These results demonstrate that the model appears biased toward predicting the minority class.

Figure 16 presents the receiver operating characteristic (ROC) curves for the FPNN and BPNN models. Both models achieved an area under the curve (AUC) value of 0.753, indicating comparable classification performance. The ROC curves are positioned above the diagonal reference line, showing that both models performed better than random classification. Overall, the results suggest that the FPNN and BPNN models demonstrated similar discriminative ability in distinguishing between the two classes.

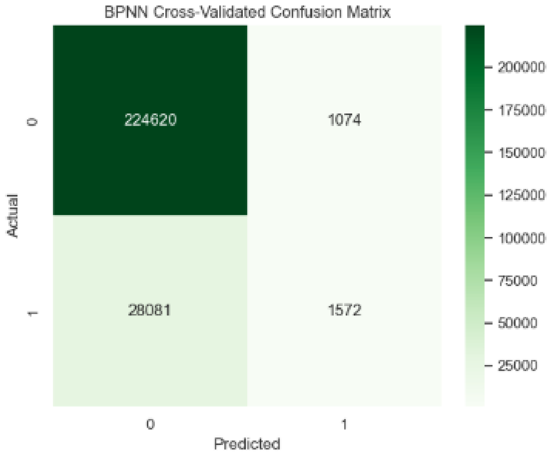


Fig 17: Forward propagation neural network cross-validated confusion matrix

From the confusion matrix in Figure 17, the following data are obtained:

- True Negatives (TN): 224,472 (class 0 correctly predicted as class 0)
- True Positives (TP): 1,794 (class 1 correctly predicted as class 1)
- False Positives (FP): 1,222 (class 0 incorrectly predicted as class 1)
- False Negatives (FN): 27,859 (class 1 incorrectly predicted as class 0)

Using the formulae provided in Section 3 of this paper, the results are presented in Table 4.

Table 4. Results of cross-validated FPNN

Metric	Value	Percentage
Accuracy	0.8861	88.61%
Precision	0.5948	59.48%
Recall	0.0605	6.05%
F1-Score	0.1100	11.00%

The cross-validated FPNN model in Table 4 achieved an accuracy of 88.61%, which is nearly identical to the BPNN model. The precision of 59.48% shows that when FPNN predicts the positive/minority class, it is correct about 59% of the time. However, the recall remains critically low at 6.05%, meaning the model captures only 6 out of every 100 actual positive cases and misses 94% of them. This leads to a low F1-Score of 11.00%, indicating a poor trade-off between precision and recall. By implication, the FPNN suffers from majority-class bias despite high accuracy. The marginally better recall and F1-Score compared to BPNN suggest FPNN is slightly more sensitive to the minority class, but the improvement is not practically significant.

Figure 18 shows the cross-validated confusion matrix for the BPNN model. The model correctly classified 224,620 samples as class 0 and 1,572 samples as class 1. It incorrectly classified 1,074 class 0 samples as class 1, while 28,081 class 1 samples were predicted as class 0. This indicates that the model performs very strongly in identifying class 0, but it has difficulty detecting class 1 samples after cross-validation.

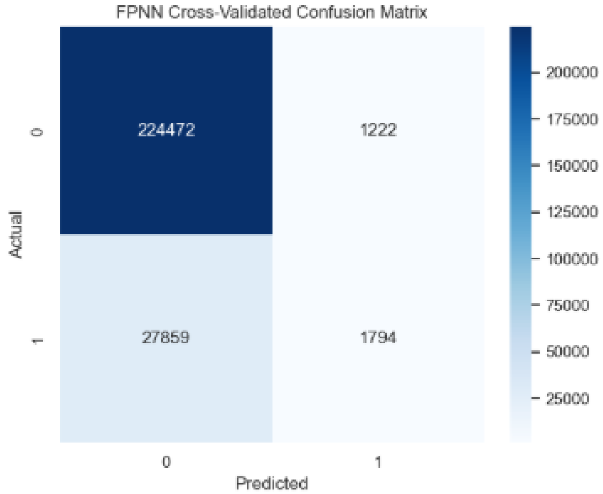


Fig 18: Backward propagation neural network cross-validated confusion matrix

From the confusion matrix in Figure 18, the following data are obtained:

- True Negatives (TN): 224,620 (class 0 correctly predicted as class 0)
- True Positives (TP): 1,572 (class 1 correctly predicted as class 1)
- False Positives (FP): 1,074 (class 0 incorrectly predicted as class 1)
- False Negatives (FN): 28,081 (class 1 incorrectly predicted as class 0)

Using the formulae provided in Section 3 of this paper, the results are presented in Table 5.

Table 5. Results of cross-validated BPNN

Metric	Value	Percentage
Accuracy	0.8858	88.58%
Precision	0.5941	59.41%
Recall	0.0530	5.30%
F1-Score	0.0973	9.73%

The cross-validated results for the BPNN model in Table 5 show an accuracy of 88.58%, which appears high at first glance. The precision of 59.41% indicates that when the model predicts the minority/positive class, it is correct about 59% of the time. More critically, the recall of only 5.30% reveals that the model identifies just 5 out of every 100 actual positive cases, meaning 95% of true positives are missed. Consequently, the F1-Score is very low at 9.73%, confirming a poor balance between precision and recall. The high accuracy but extremely low recall suggests the BPNN is biased toward predicting the majority class and fails to detect the minority class, which is often the class of interest in imbalanced problems like fraud detection.

Figure 19 presents the cross-validation accuracy of the FPNN and BPNN models across five folds. The FPNN model achieved accuracy values of approximately 0.8860, 0.8865, 0.8861, 0.8858, and 0.8862 for folds 0 to 4, respectively. Similarly, the BPNN model recorded accuracy values of approximately 0.8856, 0.8859, 0.8859, 0.8859, and 0.8859. Both models demonstrated stable accuracy across all folds, with only minor variations. Overall, the FPNN model showed slightly higher accuracy in most folds, while the BPNN model exhibited more consistent performance.

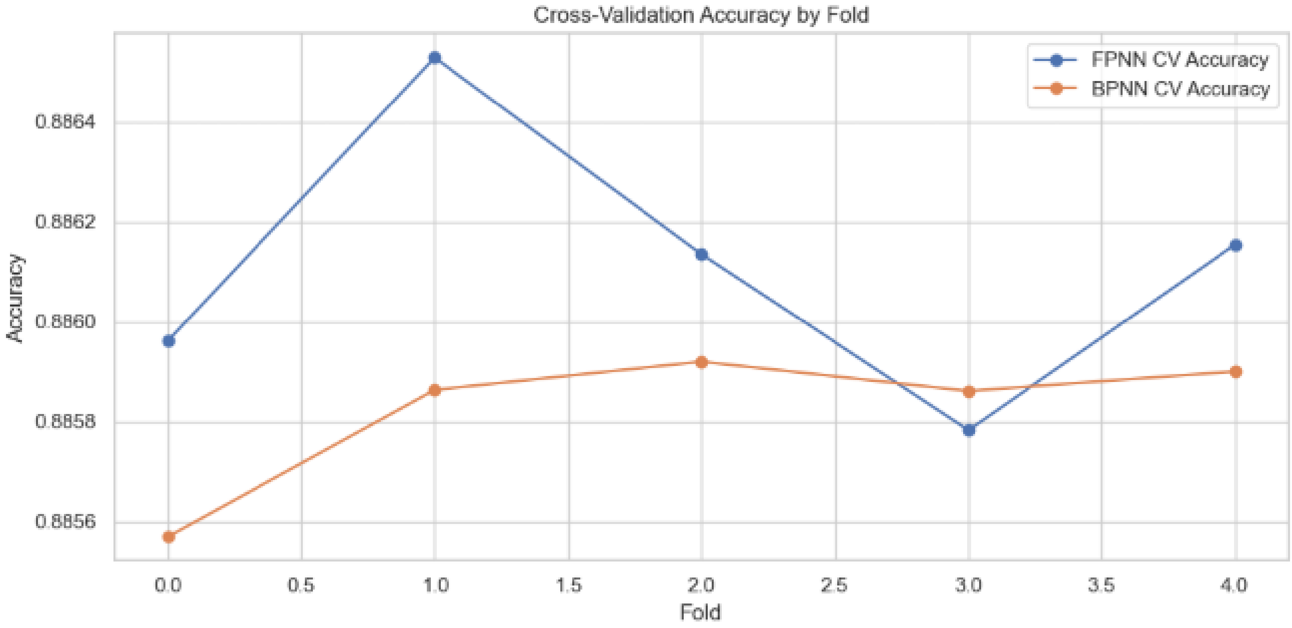


Fig 19: Cross-validation accuracy by fold (FPNN vs BPNN)

Figure 20 presents the cross-validation loss of the FPNN and BPNN models across five folds. The FPNN model recorded loss values of approximately 0.3160, 0.3136, 0.3149, 0.3139, and 0.3150 for folds 0 to 4, respectively. Similarly, the BPNN model obtained loss values of approximately 0.3156, 0.3143, 0.3147, 0.3136, and 0.3154. Both models demonstrated stable loss patterns across the folds, with only minor variations. The lowest loss for FPNN occurred at fold 1, while BPNN achieved its lowest loss at fold 3. Overall, the results indicate that both models exhibited comparable cross-validation performance, with no substantial difference in loss.

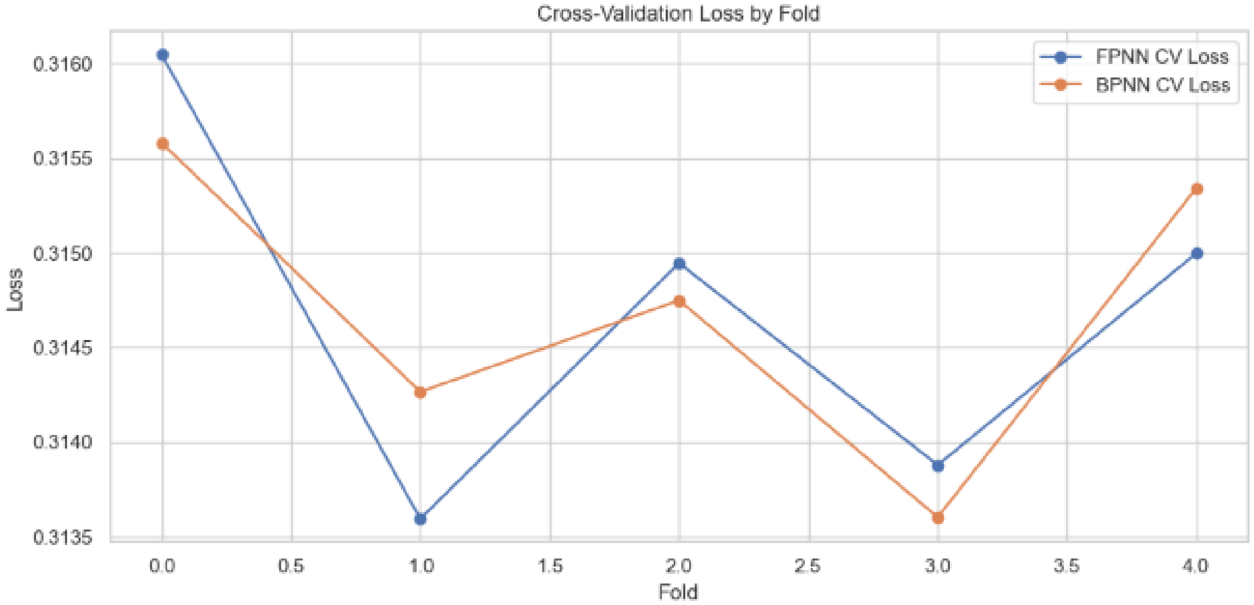


Fig 20: Cross-validation loss by fold (FPNN vs BPNN)

4.7 Results Comparison

The results comparison is split into three categories. Category A compares FPNN and BPNN before cross-validation. Category B compares FPNN and BPNN after cross-validation. Category C compares FPNN and BPNN before and after cross-validation.

Table 6. Baseline FPNN and BPNN results comparison

Model	Accuracy	Precision	Recall	F1-Score	AUC
FPNN	0.7110	0.2334	0.6513	0.3420	0.752557
BPNN	0.6920	0.2257	0.6800	0.3390	0.753332

Table 6 shows that FPNN outperforms BPNN with 71.10% vs 69.20% accuracy score, a difference of 1.90 percentage points. This suggests that FPNN correctly classifies slightly more instances overall on the baseline imbalanced dataset.

Table 7. Cross-validated FPNN and BPNN results comparison

Model	Accuracy	Precision	Recall	F1-Score	AUC
FPNN	0.8861	0.5948	0.0605	0.1100	0.752557
BPNN	0.8858	0.5941	0.0530	0.0973	0.753332

Table 7 reveals that FPNN and BPNN perform almost identically after cross-validation, with 88.61% vs 88.58% accuracy score, respectively. The 0.03 percentage point difference is negligible and suggests both models classify the majority of instances correctly.

Table 8. Before and after cross-validation results comparison

Model	Baseline Accuracy	CV Accuracy Mean	Baseline Loss	CV Loss Mean
FPNN	0.7110	0.8861	0.584308	0.314693
BPNN	0.6920	0.8858	0.605187	0.314709

The results in Table 8 show that both models improved substantially after cross-validation, especially in terms of accuracy and loss. The FPNN increased from 0.7110 baseline accuracy to 0.8861 mean cross-validated accuracy. At the same time, the BPNN improved from 0.6920 to 0.8858, indicating that cross-validation produced a more stable and reliable estimate of model performance. Similarly, the loss values dropped from 0.584308 to 0.314693 for FPNN and from 0.605187 to 0.314709 for BPNN, which suggests better learning and stronger generalization after validation across multiple folds. The comparison is also visualized in Figures 21 and 22 for accuracy and loss, respectively.

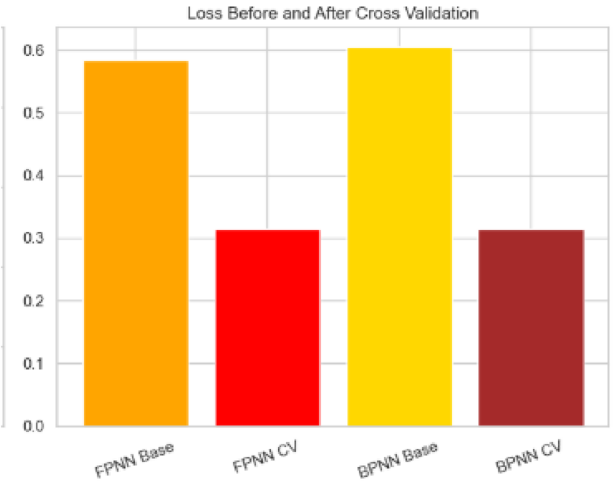


Fig 21: FPNN and BPNN accuracies before and after cross-validation

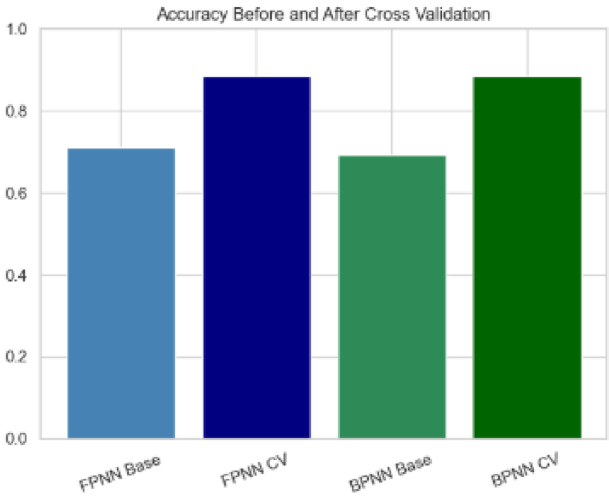


Fig 22: FPNN and BPNN loss before and after cross-validation

The McNemar test results in Table 9 show a clear difference between the two models before cross-validation, where the chi-square value is very high, and the p-value is less than 0.05.

Table 9. McNemar test results

Condition	b	c	Chi-square	p-value
Before CV	2004	1029	312.7847	0.0000
After CV	815	741	3.4248	0.0642

Table 9 indicates that the performance difference between FPNN and BPNN before cross-validation was statistically significant, meaning one model made noticeably more correct predictions than the other on the same test instances. The larger imbalance between b=2004b = 2004b=2004 and c=1029c = 1029c=1029 supports this conclusion and suggests that the models did not behave similarly on the baseline split. After cross-validation, the chi-square value dropped sharply to 3.4248, and the p-value increased to 0.0642. This implies that the performance difference between the two models was no longer statistically significant at the 5% level, so their predictive behavior became more comparable after validation across multiple folds.

Table 10. Models' comparison

Study	Methodology	Accuracy
Reference [11]	BPNN	98.01%
Reference [13]	BPNN+RF	95.1%
Reference [12]	Genetic-based BP	89%
Current Study	FPNN-Cross-Validation	88.61%
	BPNN-Cross-Validation	88.58%

Despite the fact that the models developed in this study have lower accuracy results as compared to the previous studies (see Table 10), they achieve strong accuracy results that are comparable to those found in existing research. This demonstrates the effectiveness of the FPNN and BPNN models.

5. CONCLUSION

This study compares forward and backward propagation neural network models for the prediction of loan default. Firstly, this study provides insight into the factors that significantly influence loan default, including age, interest rate, income, months employed, loan amount, employment type, cosigner, dependency, credit score, and number of credit lines. This means that younger borrowers are more likely to default on loans due to limited financial experience and stability. The likelihood of loan default can be increased by higher interest rates, because borrowers may struggle to make payments. It is less likely to default on loans by people with higher incomes, due to a stable source of income. Recently employed people are more likely to default on loans. Larger loan amounts may increase the likelihood of default, and people with unstable employment, such as those in full-time positions, tend to default on loans. Furthermore, people without a cosigner have a high probability of defaulting on a loan because a cosigner provides an additional layer of security. In addition, people with dependents are likely to default on loans because of other financial responsibilities, and people with lower credit scores are highly likely to default on loans. Lastly, borrowers with multiple credit lines may be more likely to default on loans, as they may be overextended and struggling to make payments.

Above all, this study's findings reveal that both FPNN and BPNN models are effective in predicting loan defaults and have good predictive capabilities. Future studies can build on these findings by using other datasets and incorporating hyperparameter tuning to further improve the model's performance, especially precision, to further improve the effectiveness, reliability, and practical deployability of the loan default prediction model.

6. REFERENCES

[1] S.T. Baidoo, H. Yusif, and E.K. Ayesu, "Improving loan repayment in Ghana". Journal of Management Science. 1(2), 21-42, 2020.

[2] M. Anand, A. Velu, and P. Whig, "Prediction of loan behaviour with machine learning models for secure banking". *Journal of Computer Science and Engineering (JCSE)*, *3*(1), 1-13, 2022.

[3] A.A. Egwa, B. Habeeb, A.A. Ahmad, and S.M. Bizi, "Default Prediction for Loan Lenders Using Machine Learning Algorithms", 2022.

[4] Y. Lai, "Credit Default Analysis and Prediction Based on Machine Learning". *Highlights in Business, Economics and Management*, *21*, 782-790, 2023.

[5] S. Mestiri, and S.M. Hiboun, "Credit scoring using machine learning and deep Learning-Based models", 2024.

[6] A. Alonso, and J.M. Carbó, "Understanding the performance of machine learning models to predict credit default: a novel approach for supervisory evaluation," 2021.

[7] A. Ampountolas, T. Nyarko Nde, P. Date, and C. Constantinescu, "A machine learning approach for micro-credit scoring". *Risks*, *9*(3), 50, 2021.

[8] M.H. Khedr, N.A. Azim, and A.M. Ammar, "A New Prediction Approach for Preventing Default Customers from Applying Personal Loans Using Machine Learning". *International Journal of Computer Science and Mobile Computing*, *10*(12), 71-82, 2021.

[9] M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath, "Loan default prediction using decision trees and random forest: A comparative study". In *IOP conference series: materials science and engineering* (Vol. 1022, No. 1, p. 012042). IOP Publishing, 2021.

[10] M.N. Alam, and M.M. Ali, "Loan default risk prediction using knowledge graph". In *2022 14th International Conference on Knowledge and Smart Technology (KST)* (pp. 34-39). IEEE, 2022.

[11] B. Li, "Online Loan Default Prediction Model Based on Deep Learning Neural Network". *Computational Intelligence and Neuroscience*, *2022*, 1–9, 2022.

[12] B. Chen, W. Jin, and H. Lu, "Using a genetic backpropagation neural network model for credit risk assessment in the micro, small, and medium-sized enterprises". *Heliyon*, *10*(14)., 2024.

[13] W. Sun, Y. Zhu, and Q. Hu, "Study on Credit Default Risk Prediction Model Based on BP-RF Neural Network". In *2021 2nd International Conference on Computer Science and Management Technology (ICCSMT)* (pp. 154-159). IEEE Computer Society, 2021.