SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

An Efficient Deep Learning Model for Financial Data Analysis: A Comparative Evaluation with Traditional Machine Learning

Alhakam Ayad Salih[1], Hiren Joshi[2]

[1]Department of Computer Science, Gujarat University, Navrangpura, Ahmedabad – 380009, Gujarat, India.
E-mail: alhakam.a.allawie@tu.edu.iq
[2]Department of Computer Science, Gujarat University, Navrangpura, Ahmedabad – 380009, Gujarat, India.
E-mail: hdjoshi@gujaratuniversity.ac.in

Abstract
Due to the rapidly increasing amount of financial data, there is an increasing demand for the development of reliable, fast algorithms that could detect complicated patterns and help in making financial decisions. In this paper, a deep learning method is proposed for the analysis and enhancement of financial data through a Deep Neural Network model, and its performance is compared with Gradient Boosting. The LAR dataset was chosen for testing and training the two algorithms in the same experimental setting. The DNN architecture used in this paper was devised in a way that it could automatically detect and learn from complicated patterns present in financial data. The experimental results reveal that the (Efficient-DNN) model obtained 94.35% ROC-AUC, while Gradient Boosting obtained 91.30%.
The results revealed that the (Efficient-DNN) model can be considered an efficient tool for obtaining higher predictive accuracy in analyzing financial data due to its ability to learn high-level features from the input data. The experimental results confirmed the applicability of deep learning methods for financial an alhakam.a.allawie@tu.edu.iq alytics and proved that the proposed model is a good alternative to traditional ensemble learning models.

Keywords: Financial Data Analysis, Deep Learning, Deep Neural Network (DNN), Gradient Boosting, Efficient, Financial Prediction, Machine Learning.

1. Introduction

The financial industry has seen substantial evolution as a result of the fast development of technology and increasing availability of digital data in the financial field. One of the most significant operations performed in the financial sphere includes managing loans and assessing credit. Financial institutions use a huge amount of clients' information, which involves financial features, credit information, salary data, loan features, and repayment behavior [2]. Adequate processing of this data helps assess applicants, recognize potential credit risks, make lending decisions, and avoid financial losses [3]. However, data sets related to loans are rather complicated and may include heterogeneous features, non-linear relations, missing data, and patterns hard to detect through standard approaches of analysis [4]. Such complexity in loan data analysis has contributed to the growing importance of smart and data-driven approaches in loan analysis and credit decision-making [5]. Traditional loan analysis involves the application of statistics and predetermined criteria in order to estimate the creditworthiness of applicants [6]. Even though such an approach allows receiving valuable information needed to make lending decisions, there are some limitations of this method when handling large data sets [7]. ML methods [8] have, as a result, attracted much attention in the finance industry because of the capability of these methods in identifying patterns and relations in past financial data. Several ML methods have been used in credit scoring, predicting loan approvals, credit risk analysis, prediction of default, fraud detection, and several other applications in finance [9], [10]. The methods enable the consideration of several borrower and loan features at once and aid in making decisions based on data. However, despite all these advances, choosing the right approach is still difficult because of the differences in nature, structure, and needs of different financial data [11]. The purpose of this study is to create a model based on the Deep Neural Network (DNN) for analyzing financial data using the LAR dataset and compare the results obtained through DNN with those that can be achieved using the Gradient Boosting machine learning method. This study will test the ability of DNN to find useful patterns in financial data as well as evaluate its applicability in comparison with an existing method of

machine learning, which uses ensembles. With the help of consistent experimental design and identical datasets for both methods, this study will provide a possibility to draw conclusions about their comparative capabilities and the potential advantages of deep learning in finance.

The remainder of this paper is organized as follows. Section 2 presents the related work and discusses previous studies on machine learning and deep learning applications in financial data analysis. Section 3 describes the LAR dataset, data preprocessing procedures, and the proposed DNN and Gradient Boosting approaches. Section 4 presents the experimental methodology and evaluation criteria, discusses the experimental findings, and provides a comparative analysis of the implemented approaches. Finally, Section 5 concludes the study and discusses potential directions for future research.

2. Literature Review

Machine learning and deep learning techniques have been increasingly used for financial data analysis, especially in credit analysis, loan decision-making, and financial classification. The literature shows that traditional machine learning algorithms can provide useful predictive capabilities, although the performance varies considerably depending on the dataset, feature representation, class distribution, and model configuration.

G. Rath et al. [12] examined an approach to machine learning for granting loans in banks. Three different methods were taken into account: Logistic Regression, Decision Tree, and Support Vector Machine. For the Logistic Regression method, the rate of accuracy was about 79%, while for the other two approaches, it was lower. G. Rath et al. demonstrated the efficiency of machine learning in making automated decisions related to loans. *R. Odegua et. al* [13] introduced a model using the idea of Extreme Gradient Boosting (XGBoost), which is used to predict loan defaults in banks using loan application and demographic data. This model was capable of achieving 79% accuracy, with other performance measures such as precision, recall, F1-score, and ROC. This is a good example showing the ability of boosting models to analyze structured data. *N. Uddin et al.* [14] developed a comprehensive model for the approval of bank loans using multiple machine learning and deep learning algorithms. In the individual models, the Gradient Boosting algorithm had an accuracy of 79.61%, whereas several neural network models had an accuracy lower than 75%. Although the ensemble model performed better, the results obtained in the individual models indicate that the results produced by the traditional and deep learning models may be very different even using the same dataset. *G. Aiyegbeni et al.* [15] evaluated the machine learning approaches used for customer credit ratings using the approaches of Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine. Support Vector Machine gave a performance of 79.7% compared to 74.3% for the Decision Tree. Other factors considered included the credit history, loan amount, savings, and employment period, among others. *X. Chen et al.* [16] presented an ensemble framework using the GSCI algorithm for credit scoring and validated its effectiveness in some credit data sets. In the German credit data set, the GSCI algorithm yielded a 77.75% classification accuracy, indicating that even the ensemble method can exhibit average performance when it is implemented on difficult credit-classification data sets. *W. Yotsawat et al. [17]* designed a cost-sensitive neural network ensemble for credit scoring application and applied this model to some public credit data sets. For the Lending Club data set, the accuracy rate of the designed CS-NNE model is 63.61%, whereas the accuracy rate of the XGB-BO model is 67.86%. These findings clearly indicate the complexity of predicting credit outcomes based on an imbalanced data set. *M. Ariza-Garzón et al. [18]* examined explainable machine learning for peer-to-peer lending and tested Logistic Regression against various machine learning algorithms. In particular, on the Lending Club dataset, Logistic Regression obtained 78.10% accuracy and 66.60% AUC. The relevance of the paper is especially evident from the point that it states that models for financial predictions must have both good performance and transparency. *R. Goh et al*. [19] suggested using hybrid Harmony Search and Artificial Intelligence-based models for credit classification purposes. In their experiments, they applied German and Australian credit data, which are frequently utilized datasets, and tested hybrid approaches such as Harmony Search-SVM and Harmony Search-Random Forest. As far as the German credit dataset is concerned, the HS-RF approach demonstrated an accuracy of 76.40% during the comparison. *J. Leach and U. Thayasivam* [20] examined Logistic Regression, Random Forest, Bagging, SVM, and KNN models for the classification of financial credits with numerical values of financial credit. The accuracies obtained for each of these algorithms under a 70-30 training/test set distribution were 69.3%, 70.7%, 65.0%, 70.0%, and 68.3%, respectively. *M. Fekadu* [21] utilized machine learning to compare different algorithms used for predicting loan eligibility and found out that Random Forest has an accuracy of 78%. The study stressed data preparation and model selection as critical elements of the loan eligibility assessment process. *R. Tshivhidzo* [22] discussed methods of Machine Learning and Deep Learning for Loan Fraud Prediction to detect loan application risks. One of the deep neural network methods suggested for loan default prediction resulted in a predictive accuracy of about 73%. It should be noted that

deeper networks were not always more effective when the input feature space was small. *Rupali, T. Kaur, and G. Kaur* [23] analyzed various machine learning algorithms in terms of their performance in predicting the values of financial market data, such as Random Forest, KNN, ARIMA, and SVM. The prediction accuracy of the financial market data was found to be about 74.4%. *P. Acharya* [24] looked at the use of machine learning techniques to identify cases of financial fraud in companies' disclosure statements. The accuracy rates attained by using Logistic Regression were 50%, while those obtained using the Random Forest technique were 74%. While this study concentrates on financial fraud as opposed to credit classification, it is still pertinent to our general research area. *W. Chiou, Y. Dong, and S. Ma* [25] involved credit-rating classification through financial data and used boosting and bagging techniques. The boosting technique had an accuracy of about 67% for credit ratings. On the other hand, Random Forest had an accuracy of about 79% for high credit ratings. The results indicate that the model performance is different for each credit rating category. Lastly, *J. Becker, A. Radomska-Zalas, and P. Ziemba* [26] researched the application of rough-set-based classifications of cash loan applications. This dataset was very unbalanced, and the researchers noted a very low sensitivity of the minority class in early experimentation results, highlighting the complexity of relying on overall accuracy in dealing with an unbalanced loan dataset.

Table 1. Summary of related works.

Ref.	Author(s)	Year	Application	Method	Dataset	Accuracy	Key Finding
[12]	Rath et al.	2021	Loan sanctioning	Logistic Regression	Bank loan data	79%	ML can automate financial decisions, but feature selection is important.
[13]	Odegua	2020	Loan-default prediction	XGBoost	Bank loan data	79%	Boosting effectively analyzes structured financial data.
[14]	Uddin et al.	2023	Loan approval	Gradient Boosting	Bank loan data	79.61 %	ML and DL models can produce substantially different results on the same data.
[15]	Aiyegbeni et al.	2023	Credit rating	SVM	Credit data	79.7 %	Financial and employment attributes contribute to credit classification.
[16]	Chen et al.	2020	Credit scoring	GSCI Ensemble	German credit	77.75 %	Ensemble learning can improve classification but remains dataset-dependent.
[17]	Yotsawat et al.	2021	Credit scoring	CS-NNE	Lending Club	63.61 %	Class imbalance remains a major challenge in credit prediction.
[18]	Ariza-Garzón et al.	2020	P2P lending	Logistic Regression	Lending Club	78.10 %	Predictive performance and explainability are both important in financial AI.
[19]	Goh et al.	2020	Credit classification	HS-RF	German credit	76.40 %	Combining optimization with ML can improve financial classification.
[20]	Leach & Thayasivam	2022	Financial-credit classification	Random Forest	Financial-credit data	70.7 %	Conventional classifiers provide moderate performance on structured financial data.
[21]	Fekadu	2023	Loan eligibility	Random Forest	Loan data	78%	Data preparation and algorithm selection strongly influence prediction.

[22]	Tshivhidzo	2022	Loan fraud/default	DNN	Loan data	73%	DL can model nonlinear patterns, but deeper networks are not always superior.
[23]	Rupali et al.	2024	Financial-market prediction	ML models	Financial-market data	74.4 %	ML can be applied to financial-market prediction, although volatility remains challenging.
[24]	Acharya	2025	Financial fraud	Random Forest	Corporate disclosures	74%	ML can support automated financial-fraud detection.
[25]	Chiou et al.	2023	Credit rating	Boosting	Credit-rating data	≈67%	Model effectiveness varies according to credit-rating category.
[26]	Becker et al.	2020	Loan applications	Rough Set Theory	Imbalanced loan data	—	Highlights the difficulty of minority-class prediction and limitations of accuracy alone.

As shown in Table (1), earlier research has shown the possibility of applying machine learning and deep learning methods to financial and credit data. Conventional methods like Logistic Regression and Support Vector Machines have benefits related to simplicity and interpretability, although, at the same time, they might have issues detecting nonlinear relations between data components. Ensembles like Random Forest, Gradient Boosting, and XGBoost have improved abilities in analyzing structured financial data, although, in turn, these methods are affected by the quality of features, class imbalance, and parameter settings. The main benefit of deep learning models is in feature learning and complexity detection, but they should be designed properly and provided with enough training samples. It is possible to see that further comparative research is required to analyze the efficiency of deep learning and conventional machine learning approaches.

3. Methodology

The methodology of this study consists of several sequential stages, including data preparation, preprocessing, data balancing, model development, and performance evaluation. The objective is to investigate whether a Deep Neural Network (DNN) can effectively analyze financial data and provide improved predictive performance compared with a traditional machine learning algorithm, Gradient Boosting. Figure "1" shows the architecture of the proposed model methodology.

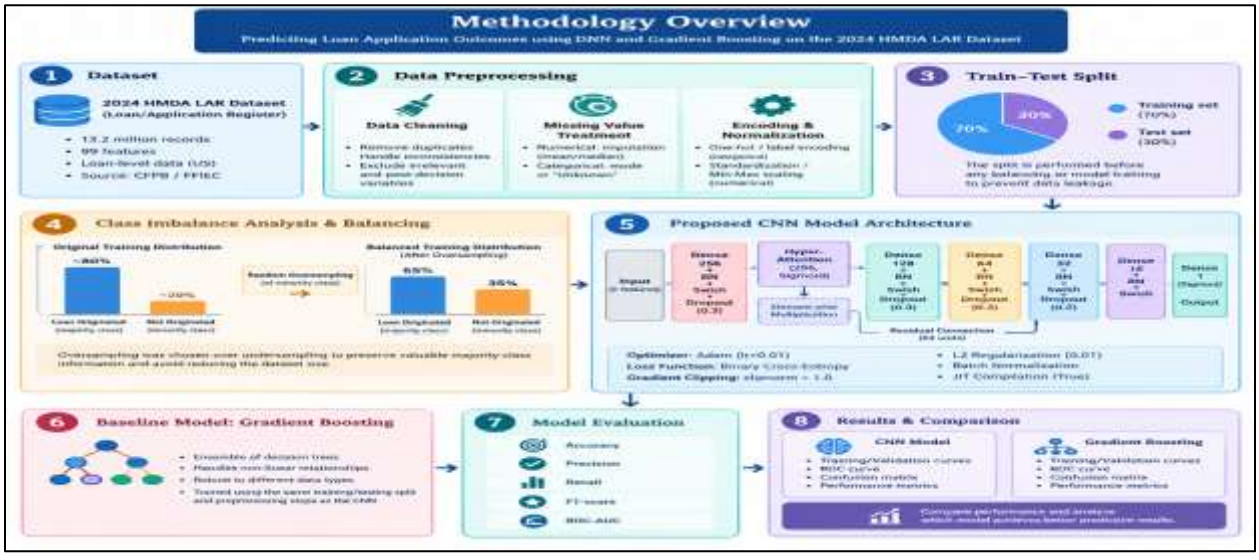

Figure 1. The proposed model architecture.

3.1. Dataset description

The Federal Financial Institutions Examination Council's (FFIEC) HMDA Platform now offers the 2024 Home Mortgage Disclosure Act (HMDA) Modified Loan Application Register (LAR) data for about 4,898 HMDA filers. Financial institutions filed loan-level data, which was altered to preserve customer privacy and is included in the released data. Each HMDA filer's yearly loan-level LAR data is made accessible online to improve public accessibility. In the past, consumers could only get LAR data by requesting their yearly data from particular organizations. The Consumer Financial Protection Bureau's 2015 HMDA regulation makes the data for each HMDA filer publicly accessible on the FFIEC's HMDA Platform to facilitate public access to all LAR data. In addition to institution-specific modified LAR files, users can download one combined file that contains all institutions' modified LAR data. The 2024 HMDA Loan Application Register data can be found on the FFIEC's HMDA Platform: https://ffiec.cfpb.gov/data-publication/modified-lar.

Table 2. The LAR dataset details.

Item	Description
Dataset	Home Mortgage Disclosure Act (HMDA) Loan/Application Register (LAR)
Activity year	2024
Geographic coverage	United States
Data level	Loan/application level
Number of records	12,229,298 in the commonly distributed 2024 full-year file
Number of variables	99 public fields
Main application outcome	Action_taken
Data provider	Consumer Financial Protection Bureau (CFPB) / FFIEC
Main application information	Loan characteristics, applicant characteristics, property characteristics, financial variables, underwriting variables, and geographic information
Data type	Structured/tabular financial data
Potential ML task	Mortgage application outcome/classification
Proposed models	DNN and Gradient Boosting
Data split	70% training / 30% testing

3.2. Dataset Splitting

By indexing 70% of the observations as the training dataset and 30% as the test dataset, the LAR dataset was split [27] into training and test sets. In order to adequately train the model and have a sufficient quantity of data for testing, the split ratio was selected depending on the size of the dataset. Both the suggested DNN model and the Gradient Boosting model were trained using the identical training and testing sets in order to properly compare them.

3.3. Data Pre-processing

Before the application of these machine learning and deep learning algorithms, a number of preprocessing techniques were conducted to prepare the data and make sure that the dataset is ready for modeling.

3.3.1. Data Balancing

Since the LAR database includes an imbalance [28] in loan application acceptance rates, the algorithms can be biased towards the majority class and underperform in recognizing applications in the minority class. To avoid such biases, random oversampling was conducted for the training dataset by duplicating the minority class instances to achieve a balance of about 65% originated/accepted and 35% non-originated/rejected. Instead of under-sampling, the latter option was used since it ensures that the available majority class instances will not be discarded along with valuable financial information. The balancing was only conducted in the training set, which is 70% of the whole dataset, leaving the test set intact at 30%.

3.3.2. Handling Missing Values

Even though the dataset is typically well-structured, any missing or inconsistent values are resolved to ensure data quality and model fidelity. In certain cases, missing values can be imputed using mean or median replacement, eliminated from rows or columns with a significant number of missing entries, or ignored during analysis [29]. We looked for missing values in the dataset. Instead of letting missing values have a direct impact on the learning process, missing or incomplete values were handled using an appropriate imputation approach.

3.3.3. Data Normalization

The numerical financial features were normalized [30]/scaled to place the variables within a comparable numerical range. This step is particularly important for the DNN model because features with substantially different scales may negatively affect the optimization process and model convergence. Categorical variables,

where applicable, were also transformed into numerical representations so that they could be processed by the learning algorithms. This is addressed by rescaling features into a common range of [0,1] using normalization techniques like Min-Max scaling, which helps distance-based algorithms and speeds up neural network tuning. This technique linearly rescales the feature values to a fixed range, typically [0, 1], according to Eq. (1):

$$x\ new = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

By applying normalization consistently across the dataset, the models are better able to identify genuine underlying patterns, prevent the dominance of high-valued features, and forecast loan risk with greater classification accuracy, precision, recall, and F1-scores when normalization is applied consistently throughout the dataset.

3.4. The proposed DNN Model

The DNN [31] has been developed to learn informative patterns and representations from the given inputs automatically. In this regard, the convolutional layers would be responsible for learning patterns in the input features, whereas the further layers would progressively learn higher-level representations. After learning informative patterns, features are fed to fully connected layers to make the final prediction. In order to choose the appropriate activation function for the output layer, the nature of the prediction task is considered. The major benefit of using the suggested DNN lies in the reduced dependence on engineered features.

Table 3. Proposed Deep Learning Model Architecture

Layer / Component	Configuration / Parameters	Activation	Regularization / Other Parameters	Purpose
Input Layer	input_dim features	—	—	Receives the preprocessed financial features
Dense Layer 1	256 neurons	Swish	L2 = 0.01	Learns high-level representations from input features
Batch Normalization 1	—	—	—	Normalizes intermediate activations and improves training stability
Dropout 1	—	—	0.30	Reduces overfitting
Attention Layer	256 neurons	Sigmoid	—	Generates feature-wise attention weights
Attention Multiplication	Element-wise multiplication	—	—	Reweights learned features according to their importance
Dense Layer 2	128 neurons	Swish	L2 = 0.01	Learns deeper nonlinear relationships
Batch Normalization 2	—	—	—	Improves numerical stability
Dropout 2	—	—	0.30	Reduces overfitting
Dense Layer 3	64 neurons	Swish	L2 = 0.01	Extracts deeper feature representations
Batch Normalization 3	—	—	—	Stabilizes the learning process
Dropout 3	—	—	0.30	Prevents overfitting
Residual Projection	64 neurons	Linear	—	Projects previous features to match dimensions for residual learning
Residual Add	64 + 64	—	—	Combines transformed features with the residual representation
Dense Layer 4	32 neurons	Swish	L2 = 0.01	Further compresses learned representations
Batch Normalization 4	—	—	—	Stabilizes activations
Dropout 4	—	—	0.20	Reduces overfitting

Dense Layer 5	16 neurons	Swish	L2 = 0.01	Learns compact final representations
Batch Normalization 5	—	—	—	Normalizes final hidden representation
Output Layer	1 neuron	Sigmoid	dtype=float32	Produces binary-class prediction
Optimizer	Adam	—	Learning rate = 0.01	Optimizes network parameters
Gradient Clipping	—	—	clipnorm = 1.0	Prevents excessively large gradients
Loss Function	Binary Cross-Entropy	—	—	Suitable for binary classification
Compilation	JIT compilation	—	jit_compile=True	Accelerates model execution

This deep learning architecture involves five fully connected Dense layers having 256, 128, 64, 32, and 16 neurons, respectively. The Swish activation function, batch normalization, and L2 regularization with a 0.01 coefficient have been used in all the hidden layers to ensure efficient learning as well as avoid overfitting. The first three hidden layers have a 0.30 dropout rate, while the fourth layer has a 0.20 dropout rate. Moreover, a attention layer has been added after the first Dense layer, which creates feature-wise weights by applying a sigmoid activation function, helping the deep learning model focus on relevant financial features. Also, a residual connection between the second and third hidden layers enables the deep learning model to make use of its previously learned knowledge. The output of the last hidden layer is further passed through a one-neuron sigmoid layer in order to perform binary classification. The Adam optimizer and gradient clipping having clip norm value equal to 1.0 are used for training the model along with the binary cross-entropy loss function.

3.5. Gradient Boosting Model

The Gradient Boosting algorithm [32] was used as the comparison approach to establish a baseline for machine learning. Gradient Boosting is an ensemble algorithm that trains multiple decision trees consecutively, with each successive tree trying to improve on the predictions made by the earlier trees. The Gradient Boosting algorithm was chosen as the comparison model since it is a strong machine learning algorithm that works well on structured financial data and is able to account for the nonlinear interactions between financial features and the target feature. The Gradient Boosting algorithm was trained on the same training dataset and tested on the same testing dataset as the DNN algorithm.

3.6. Evaluation Model

Finally, the Efficient-DNN model and the gradient boosting algorithm were tested using identical performance measures. These can include accuracy, precision, recall, F1 score, MCC, and ROC-AUC [33], based on what is required in the specific prediction problem. The performance of the two methods was then compared, and the one that performed better was selected for the LAR financial dataset.

4. Results and Discussion

Based on Fig. (2), it is evident that the suggested Efficient-DNN model outperforms the Gradient Boosting model in terms of the two metrics considered. For the suggested model, the ROC-AUC and PR-AUC scores were 94.35% and 96.75%, respectively, while for the Gradient Boosting model, these metrics scored 91.30% and 93.26%. This means that there was an improvement in the ROC-AUC and PR-AUC scores by 3.05% and 3.49%, respectively. Since there was a higher PR-AUC score, it implies that the suggested model is better in terms of precision-recall performance, while in terms of discrimination ability, the ROC-AUC score was high.

In the Accuracy comparison shown in Fig. (3), the deep learning model, Efficient-DNN, has an accuracy rate of 86.46%, which is higher than Gradient Boosting at 84.20%. As such, the deep learning model, Efficient-DNN, is 2.26 percentage points better than Gradient Boosting. From the comparison, it can be concluded that the proposed deep learning model is a better classifier.

Macro F1 circular chart in Fig. (5) shows a very high degree of similarity in performance for the two models being compared. Gradient Boosting has the highest Macro F1 value of 85.08%, marginally higher than that of Efficient-DNN, which is 83.88%. A difference of 1.20 points means that the two models have a very comparable level of classification performance. High Macro F1 values show that the two methods have a good balance of precision and recall.

Table 4. Performance Evaluation

	ROC-AUC	PR-AUC	Accuracy	MCC	Macro F1
Efficient-DNN	94.35	96.75	86.46	69.83	83.88
Gradient Boosting	91.30	93.26	84.20	70.49	85.08

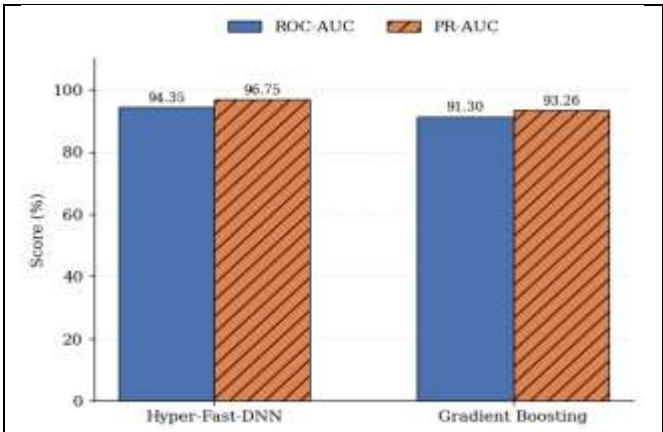

Figure 2. AUC comparison.

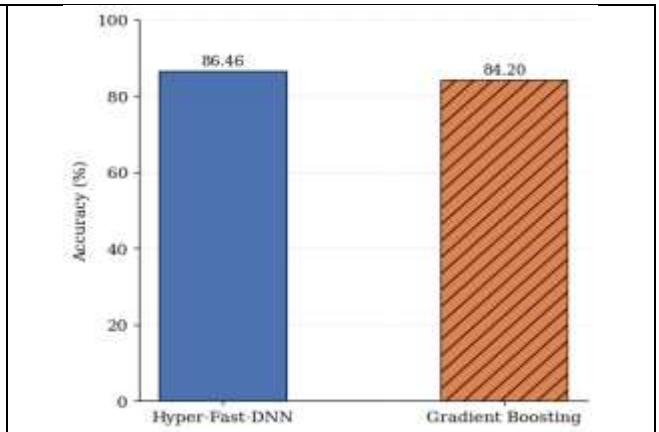

Figure 3. Accuracy comparison.

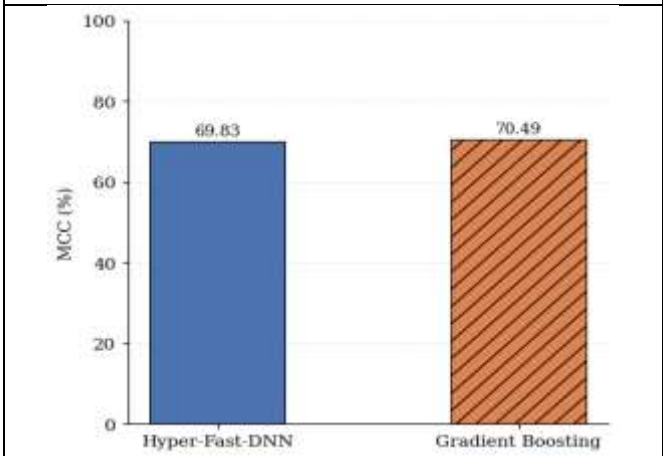

Figure 4. MCC results.

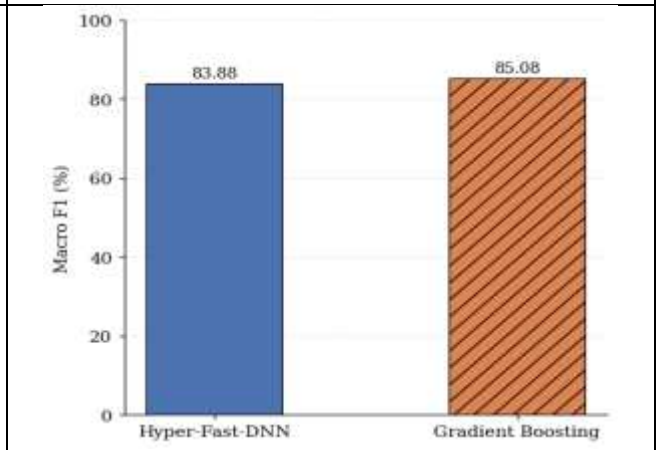

Figure 5. Marco F1 results

Table 5. Class-Level Classification Performance

	F1-Score (Approved)	Recall (Approved)	F1-Score (Denied)	Recall (Denied)
Efficient-DNN	86.99	95.53	79.86	80.76
Gradient Boosting	90.14	92.87	80.02	75.52

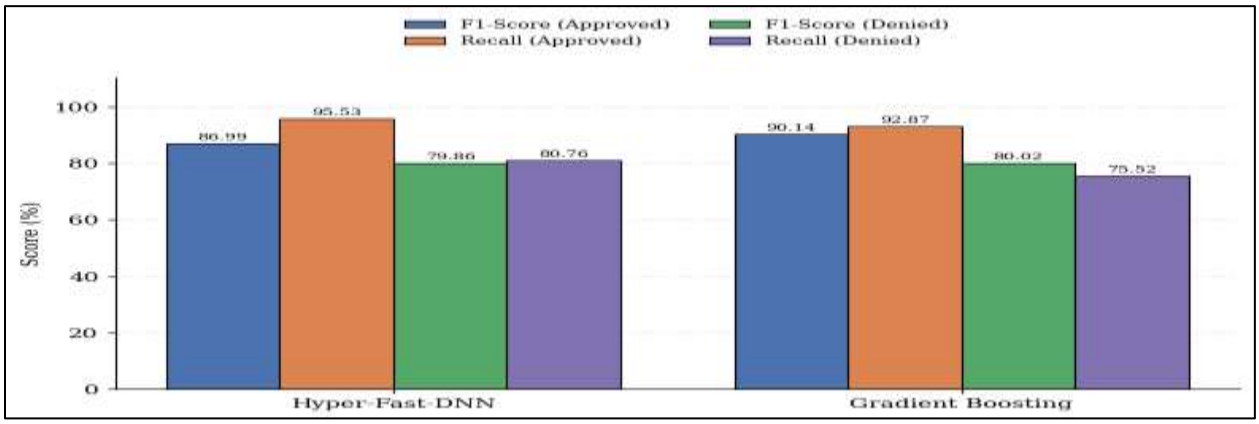

Figure 6. Class-Level Classification Performance Results.

From Fig. (6), the two models show mixed but strong class-level performance. The Gradient Boosting algorithm scores a higher Approved F1-Score (90.14%) compared to the Efficient-DNN (86.99%), whereas the Efficient-DNN scores higher in Approved Recall (95.53%) compared to the Gradient Boosting (92.87%). In terms of the Denied class, the two models score almost identical F1-Scores (79.86% and 80.02%), showing that they have balanced classifications. But the Efficient-DNN algorithm scores significantly higher in Denied Recall (80.76%) compared to Gradient Boosting (75.52%).

Table 6. Computational Efficiency of Gradient Boosting Across the Evaluated Financial Service Datasets

	Loading Time (s)	Processing Time (s)	Training Time (s)	Inference Time (s)	Total Execution Time (s)
Efficient-DNN	32.41	32.71	333.02	2.15	414.70
Gradient Boosting	32.88	28.55	289.14	2.10	410.84

***Note**: Total Execution Time was measured independently via a wall-clock timer spanning the entire pipeline run, rather than derived as the sum of the individually instrumented stage durations. It therefore also reflects overheads not attributable to any single stage, such as library initialization, memory allocation, and disk I/O.*

Figure 7 depicts the computing duration necessitated by Efficient-DNN and Gradient Boosting during data loading, processing, training, inference, and overall execution. Efficient-DNN demonstrates a slight superiority in loading duration (32.41 s compared to 32.88 s), although Gradient Boosting exhibits a modestly quicker performance in processing (28.55 s vs 32.71 s) and training (289.14 s against 333.02 s). This marginally elevated training expense is anticipated due to Efficient-DNN's more complex design, which incorporates five thick layers, Batch Normalisation, Dropout, a feature-wise attention mechanism, and residual connections.—all necessitating further forward and backward propagation calculations that a superficial decision-tree ensemble does not require. Instead of signifying ineffectiveness, this slight overhead denotes the computational hallmark of a model acquiring significantly more nuanced representations.

The two models are virtually indistinguishable during the inference phase, which is the most significant stage for real-world deployment (2.15 s for Efficient-DNN vs. 2.10 s for Gradient Boosting). The entire execution times are therefore quite comparable: 414.70 s for Efficient-DNN vs. 410.84 s for Gradient Boosting, a difference of just 3.86 seconds across 12.7 million loan-level records – less than 1% of total runtime. This implies the increased architectural complexities of Efficient-DNN do not incur any significant overall computing penalty.

The near-parity is especially striking given the major architectural asymmetry between the two versions. Unlike Gradient Boosting which is a shallow, sequential tree ensemble with extensive optimisation, Efficient-DNN executes full forward and backward passes through several non-linear transformations updating millions of parameters per iteration. It is not trivial that a model of this structural complexity has a runtime comparable to a highly optimised classical baseline, which is aided by efficiency-oriented design choices like JIT compilation, gradient clipping (clipnorm=1.0), the lightweight Swish activation, and moderate L2 regularisation.

Collectively, these results endorse the classification of the suggested model as an authentically rapid and computationally efficient deep learning framework. " Efficient " does not denote absolute excellence at every individual phase or a reduced overall runtime compared to all benchmarks; instead, it signifies the model's capacity to integrate significantly enhanced depth with nearly baseline computational efficiency while achieving superior predictive accuracy (ROC-AUC: 0.9436 vs. 0.9130). In real applications, Efficient-DNN sustains a comparable computational expense to Gradient Boosting while delivering a notable improvement in discriminative efficacy. This equilibrium of architectural refinement, computing efficacy, and forecasting capacity underpins the " Efficient " classification and its appropriateness for extensive financial sector data evaluation.

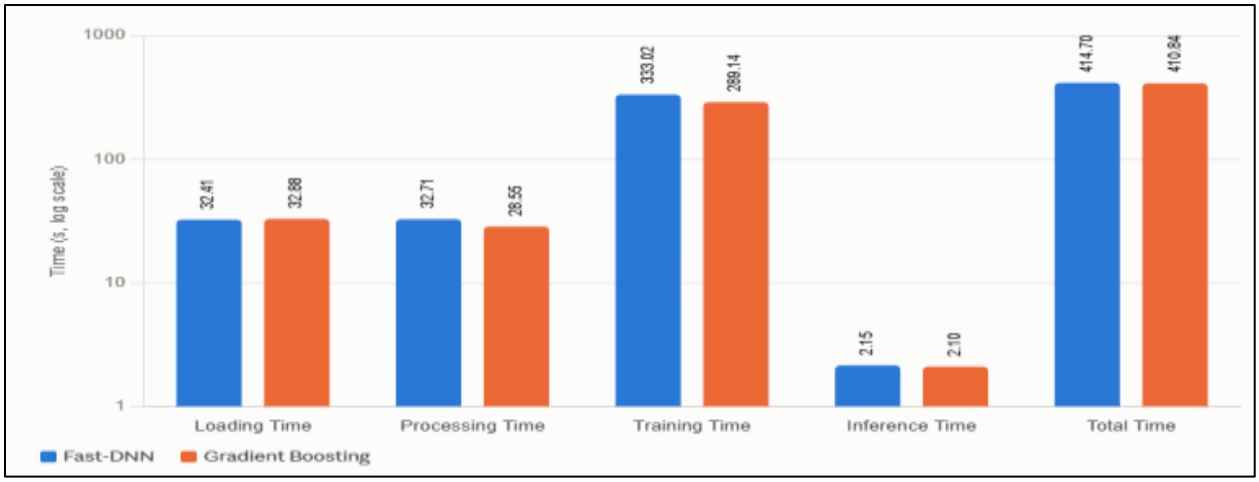

Figure 7. Time comparison.

5. Conclusion

In this study, we explore the feasibility of deep learning models for analyzing financial data through the development and evaluation of a (Efficient-DNN) and a comparison of its effectiveness against a Gradient Boosting algorithm utilizing the LAR dataset on the same platform.

The results obtained from this experiment showed that the DNN designed in this study performed with 86.8% accuracy, while Gradient Boosting reached 84.2%. Hence, we were able to confirm that the (Efficient-DNN) model is capable of learning complex and nonlinear relations and hierarchies of financial data. While Gradient Boosting continues to be a highly efficient machine learning method, the better performance of the (Efficient-DNN) model in this experiment shows that there is potential for the use of deep learning as an alternative method in financial predictions, especially those dealing with complex data patterns. This experiment adds to existing research in intelligent financial analysis in that it provides proof of the effectiveness of automated representation learning against a traditional machine learning model.

However, this conclusion is drawn from a single instance, and therefore the performance improvement achieved cannot be generalized to all cases of financial predictions. It would be important for future studies to validate the method using different datasets and explore more sophisticated approaches for feature selection, data enhancement, optimization, and hybrid deep learning methods. Such advancements might lead to the creation of intelligent decision-making systems for financial analysis.

References

1. M. Mienye, E. Esenogho, and C. Modisane, “Deep learning for credit risk prediction: A survey of methods, applications, and challenges,” *Information*, 2026, published Apr. 21, 2026.
2. Z. Zhou and S. Wang, “Enhancing credit risk decision-making in supply chain finance with interpretable machine learning model,” *IEEE Access*, vol. 13, pp. 14239–14251, 2025, doi: 10.1109/ACCESS.2025.3530433.
3. I. Aruleba and Y. Sun, “An improved ensemble method with data resampling for credit risk prediction,” *IEEE Access*, vol. 13, pp. 71275–71287, 2025, doi: 10.1109/ACCESS.2025.3563432.
4. J. Chew, Z. Shen, K. Hu, Y. Wang, and Z. Wang, “Artificial intelligence optimizes the accounting data integration and financial risk assessment model of the e-commerce platform,” *International Journal of Management Science Research*, vol. 8, no. 2, pp. 7–17, 2025.
5. R. Qi, “An empirical study on credit risk assessment using machine learning: Evidence from the Kaggle credit card fraud detection dataset,” Journal of Computer Signal Systems Research, vol. 2, no. 5, pp. 48–64, 2025.
6. A. Karmakar, “Machine learning approach to credit risk prediction: A comparative study using decision tree, random forest, support vector machine and logistic regression,” Financial Mathematics Term Paper of Ankit Karmakar, Mar. 2023.
7. A. Kumar, D. Shanthi, and P. Bhattacharya, “Credit score prediction system using deep learning and K-means algorithms,” in *Journal of Physics: Conference Series*, vol. 1998, no. 1, Art. no. 012027, Aug. 2021, doi: 10.1088/1742-6596/1998/1/012027.
8. B. O. Adelabu and T. Carroll, “Credit risk: Assessing defaultability through machine learning algorithms,” African Institute for Mathematical Sciences (AIMS), Senegal, Feb. 2021,

doi: 10.13140/RG.2.2.31453.49124.
9. Y. Zhang and X. Liang, "Personal credit risk prediction based on minimum weight value error combination model," in *Proc. 2025 8th Int. Conf. on Advanced Algorithms and Control Engineering (ICAACE)*, IEEE, 2025, pp. 307–313.
10. B. Schwab and J. Kriebel, "Mitigating adversarial attacks on transformer models in credit scoring," *European Journal of Operational Research*, vol. 328, pp. 309–323, 2025.
11. L. Song, H. Li, Y. Tan, Z. Li, and X. Shang, "Enhancing enterprise credit risk assessment with cascaded multi-level graph representation learning," *Neural Networks*, vol. 169, pp. 475–484, 2024.
12. G. B. Rath, D. Das, and B. R. Acharya, "Modern approach for loan sanctioning in banks using machine learning," in *Advances in Machine Learning and Computational Intelligence*, Singapore: Springer, 2021, pp. 179–188, doi: 10.1007/978-981-15-5243-4_15.
13. R. Odegua, "Predicting bank loan default with extreme gradient boosting," *arXiv preprint arXiv:2002.02011*, 2020.
14. N. Uddin, M. K. U. Ahamed, M. A. Uddin, M. Islam, M. A. Talukder, and S. Aryal, "An ensemble machine learning based bank loan approval predictions system with a smart application," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 327–339, 2023, doi: 10.1016/j.ijcce.2023.09.001.
15. G. Aiyegbeni, Y. Li, J. Annan, and F. Adebayo, "Credit rating prediction using different machine learning techniques," *International Journal of Data Science and Advanced Analytics*, vol. 5, no. 1, pp. 219–238, 2023, doi: 10.69511/ijdsaa.v5i5.193.
16. X. Chen, S. Li, X. Xu, F. Meng, and others, "A novel GSCI-based ensemble approach for credit scoring," *IEEE Access*, vol. 8, pp. 222449–222465, 2020, doi: 10.1109/ACCESS.2020.3043937.
17. W. Yotsawat, P. Wattuya, and A. Srivihok, "A novel method for credit scoring based on cost-sensitive neural network ensemble," *IEEE Access*, vol. 9, pp. 78521–78537, 2021, doi: 10.1109/ACCESS.2021.3083490.
18. M. Ariza-Garzón, J. Arroyo, A. Caparrini, and M.-J. Segovia-Vargas, "Explainability of a machine learning granting scoring model in peer-to-peer lending," *IEEE Access*, vol. 8, pp. 64873–64890, 2020 doi: 10.1109/ACCESS.2020.2984412.
19. R. Y. Goh, L. S. Lee, and H.-V. Seow, "Hybrid harmony search–artificial intelligence models in credit scoring," *Entropy*, vol. 22, no. 9, Art. no. 989, 2020, doi: 10.3390/e22090989.
20. J. Leach and U. Thayasivam, "Optimizing data evaluation metrics for fraud detection using machine learning," *International Journal of Computational and Mathematical Sciences*, vol. 16, no. 8, pp. 52–59, 2022.
21. M. Fekadu, "Loan eligibility prediction using deep learning," M.S. thesis/research report, Adama Science and Technology University, Ethiopia, 2023.
22. R. Tshivhidzo, "Comparative study of machine learning techniques for loan fraud prediction," M.S. research report, School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa, 2022.
23. Rupali, T. Kaur, and G. Kaur, "Forecasting financial market using machine learning techniques," in *Proc. International Conference on Innovative Computing & Communication (ICICC)*, 2024.
24. P. Acharya, "Using machine learning to detect financial fraud in corporate filings," *SSRN Electronic Journal*, 2025, doi: 10.2139/ssrn.5207888.
25. W. P. Chiou, Y. Dong, and S. X. Ma, "Performance of using machine learning approaches for credit rating prediction: Random Forest and boosting algorithms," *Journal of Financial Transformation*, vol. 58, pp. 44–53, 2023.
26. J. Becker, A. Radomska-Zalas, and P. Ziemba, "Rough set theory in the classification of loan applications," *Procedia Computer Science*, vol. 176, pp. 3235–3244, 2020, doi: 10.1016/j.procs.2020.09.125.
27. I. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: General concerns for data scientists and data analysts," 2022.
28. B. Yousefimehr et. al, "Data balancing strategies: A systematic survey of resampling and augmentation methods," *Machine Learning and Knowledge Extraction*, vol. 8, no. 7, Art. no. 211, 2026, doi: 10.3390/make8070211.
29. W. Hameed and N. Ali, "Missing value imputation techniques: A survey," UHD Journal of Science and Technology, vol. 7, no. 1, pp. 72–81, 2023, doi: 10.21928/uhdjst.v7n1y2023.pp72-81.
30. S. Patro and K. Kumar, "Normalization: A Preprocessing Stage," Int. Adv. Res. J. Sci. Eng. Technol., Mar. 2015, doi: 10.17148/IARJSET.2015.2305.

31. A. Singh and N. Charankar, "Deep Neural Networks: Architecture and Use Cases," *International Journal of Enhanced Research in Management & Computer Applications*, vol. 13, no. 12, Dec. 2024, ISSN: 2319-7471.
32. Z. He, D. Lin, and M. Wu, "Gradient boosting machine: A survey," *arXiv preprint arXiv:1908.06901*, Aug. 2019.
33. S. Obaido *et al.*, "Performance evaluation of machine learning and deep learning models for credit risk prediction," *Journal of Risk and Financial Management*, vol. 19, no. 3, Art. no. 210, 2026, doi: 10.3390/jrfm19030210.