IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 October 2024, accepted 27 November 2024, date of publication 2 December 2024,
date of current version 30 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3509774

RESEARCH ARTICLE

Machine Learning and Deep Learning for Loan Prediction in Banking: Exploring Ensemble Methods and Data Balancing

ESLAM HUSSEIN SAYED[1,2], AMERAH ALABRAH[3], KAMEL HUSSEIN RAHOUMA[4], (Member, IEEE), MUHAMMAD ZOHAIB[5], AND RASHA M. BADRY[1]

[1]Information System Department, Faculty of Computers and Information, Fayoum University, Faiyum 2933110, Egypt
[2]Information System Department, Faculty of Computer Science, Nahda University, Beni Suef 62764, Egypt
[3]Department of Information Systems, College of Computer and Information Science, King Saud University, Riyadh 11543, Saudi Arabia
[4]Electrical Engineering Department, Faculty of Engineering, Minia University, Minya 2431436, Egypt
[5]Software Engineering Department, Lappeenranta-Lahti University of Technology, 53851 Lappeenranta, Finland

Corresponding authors: Eslam Hussein Sayed (eslaamhusseinn@gmail.com) and Amerah Alabrah (aalobrah@ksu.edu.sa)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project, under Grant RSP2024R244.

ABSTRACT The prediction of loan defaults is crucial for banks and financial institutions due to its impact on earnings, and it also plays a significant role in shaping credit scores. This task is a challenging one, and as the demand for loans increases, so does the number of applications. Traditional methods of checking eligibility are time-consuming and laborious, and they may not always accurately identify suitable loan recipients. As a result, some applicants may default on their loans, causing financial losses for banks. Artificial Intelligence, using Machine Learning and Deep Learning techniques, can provide a more efficient solution. These techniques can use various classification algorithms to predict which applicants will likely be eligible for loans. This study uses five Machine Learning classification algorithms (Gaussian Naive Bayes, AdaBoost, Gradient Boosting, K Neighbors Classifier, Decision Trees, Random Forest, and Logistic Regression) and eight Deep Learning algorithms (MLP, CNN, LSTM, Transformer, GRU, Autoencoder, ResNet, and DenseNet). The use of Ensemble Methods and SMOTE with SMOTE-TOMEK Techniques also has a positive impact on the results. Four metrics are used to evaluate the effectiveness of these algorithms: accuracy, precision, recall, and F1-measure. The study found that DenseNet and ResNet were the most accurate predictive models. These findings highlight the potential of predictive modeling in identifying credit disapproval among vulnerable consumers in a sea of loan applications.

INDEX TERMS Customer loan prediction, artificial intelligence, data preprocessing, model optimization, machine learning, deep learning, classification models.

I. INTRODUCTION

The credit-lending sector in the banking industry has seen significant expansion and increased competition due to the rise of numerous credit start-ups. This growth has led to a higher number of loan applications and spending, which in turn has caused a rise in losses associated with poor credit. Banks and financial institutions offer credit loans that require repayment within a specified period, with or without interest, for various purposes such as consumer expenses, education, medical expenses, travel, and business ventures. To manage the risk of declining credit quality caused by the increase in loan applications and intensifying competition, banks and financial institutions should use this research to create effective models that utilize available data and build robust predictive frameworks [1]. Loans play a vital role in the core operations of almost every bank, with a substantial portion of a bank's resources derived from the profits generated through lending.

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

Within the banking system, the primary objective is to allocate these resources to trustworthy individuals or entities. Despite the rigorous verification and validation processes implemented by many banks and financial institutions, there remains uncertainty regarding whether the selected applicant is the most deserving among the available candidates [2]. By utilizing this method, we can assess the reliability of an individual applicant, automating the validation of attributes through the use of Artificial Intelligence techniques. Financial institutions and banks manage a variety of loan types, including education, business, home, and personal loans, among others. These institutions have a presence in diverse locations, from villages to towns and cities. When a customer requests a loan, these entities must verify the customer's information to determine loan eligibility. The loan assessment process is prone to errors, with two primary types: Type 1 error occurs when a loan officer mistakenly identifies a potentially good loan as bad, leading to its rejection and impacting customer relations. Conversely, Type 2 error happens when a loan officer inaccurately categorizes a potentially bad loan as good, resulting in approval but leading to defaults and financial losses for the institution. The loan assessment process is susceptible to two primary types of errors, each with significant implications for both customers and financial institutions; Type 1 Error: False Negatives occurs when a loan officer mistakenly identifies a potentially good loan as bad, leading to the rejection of an application that, in reality, may have been profitable and low-risk, impacting loan decisions through damaged customer relations, revenue loss, and decreased market competitiveness, while Type 2 Error: False Positives occurs when a loan officer inaccurately categorizes a potentially bad loan as good, resulting in the approval of a high-risk loan, leading to potential defaults and financial losses, impacting loan decisions through increased financial risk, inadequate risk management, and potential regulatory scrutiny, and to enhance the accuracy of loan assessments and improve decision-making, it is essential to balance the trade-offs between Type 1 and Type 2 errors by implementing advanced data preprocessing and machine learning techniques such as ensemble methods, advanced sampling techniques, and interpretability techniques.Therefore, loan officers aim to adopt strategies that reduce errors in their assessments [3]. The significance of credit rating in today's financial landscape has grown critical, drawing the attention of researchers due to advancements in data science and AI. The emphasis has been on loan prediction and credit risk assessment, with the increasing demand for loans necessitating more credit scores and loan prediction models. Over time, various techniques have been used to calculate individual credit scores, with extensive research conducted. In contrast to the past, when creditworthiness was mainly based on expert opinions and subjective judgments, automated approaches are now preferred. Machine learning algorithms and neural networks have long been advocated for by scientists and banking professionals to determine credit scores and assess risks. Notable progress has been made, offering opportunities for further exploration and analysis to deepen our understanding in this field [1]. To ensure effectiveness, it's vital to develop and verify machine learning models capable of recognizing influential patterns and indicators related to loan defaults. The choice of model is crucial as it significantly impacts the prediction system's efficiency and accuracy. Various models have been employed to predict loan defaults, with some performing better than others, and the main goal is to decide whether an applicant's loan should be approved based on well-trained models. [2]. This paper evaluates various AI techniques to minimize errors and enhance the precision of loan assessments. By utilizing a wide array of machine learning classification methods, the paper examines a bank loan dataset to pinpoint significant features and variables potentially affecting loan repayment [1]. Furthermore, we possess data on the evaluation statistics of our models, including Accuracy, Recall, Precision, AUC-ROC, MCC and F1 Score. The aim of this paper is to implement a swift, efficient, and straightforward approach to identify deserving individuals. the objectives of the study regarding gaps in the literature and the unique contributions of advanced data preprocessing and AI techniques in loan prediction research.

Identified Gaps in the Literature

1. Underutilization of Ensemble Methods: While there is an abundance of studies exploring individual Machine Learning algorithms for loan prediction, there remains a significant gap in research focused on ensemble methods. This study aims to address this gap by systematically evaluating the performance of ensemble approaches, such as Random Forest and Gradient Boosting, to determine their effectiveness in enhancing predictive accuracy in comparison to standalone models.

2. Insufficient Handling of Imbalanced Datasets: Many existing studies recognize the challenge of class imbalance in credit risk assessments but often employ only basic techniques to mitigate this issue. This research specifically targets this gap by using advanced methods such as SMOTE and SMOTE-TOMEK. By applying these techniques, we aim to improve model performance on minority classes, ultimately leading to more balanced and reliable predictions.

3. Lack of Focus on Model Interpretability: The growing need for transparency in credit risk assessments introduces another gap. While many studies aim for high accuracy, they often overlook the interpretability of Machine Learning models. This study emphasizes the inclusion of interpretability techniques such as SHAP and LIME, thereby addressing the critical need for transparency and enhancing stakeholder trust in the predictive models. The structure of this paper is as follows: Following the introduction, the second section provides a literature review, and section three presents the problem statement. Section four delves into the research methodology, while section five showcases the results and discussion of our case paper, culminating in section six, which emphasizes the conclusions.

II. BACKGROUND

Many authors have utilized various ML techniques for predicting loans for banking customers. In [1], The primary focus of the authors was on utilizing machine learning models to predict loan behavior for secure banking. They utilized the Extra Trees Classifier algorithm, renowned for its high efficacy in achieving elevated accuracy levels. The classifier accuracies were as follows: Random Forest Classifier (85.55%), Cat Boost Classifier (84.92%), Light Gradient Boosting (84.49%), and Extreme Gradient Boosting (83.87%). Remarkably, the Extra Trees Classifier excelled over the others, achieving an accuracy of 86.17%. In [2], The authors mainly focused on predicting customer loan eligibility using supervised learning techniques. They employed five models: Random Forest, Logistic Regression, Decision Tree, KNN, and SVM. Accuracy, a widely used metric in machine learning algorithms to assess model performance, yielded the following percentages for the machine learning models: Random Forest achieved 82%, Logistic Regression achieved 73%, Decision Tree achieved 72%, KNN achieved 59%, and SVM achieved 78%. The Random Forest technique particularly excelled, demonstrating a significantly enhanced accuracy of 82%. In [3], The authors primarily concentrated on assessing consumer loans using machine learning techniques. They utilized three models: SVM, Decision Tree, and AdaBoost. The results indicated that the accuracy achieved for the ML models was as follows: SVM (77.30%), Decision Tree (78.70%), and AdaBoost (74.5%). The Decision Tree technique particularly distinguished itself as highly effective, demonstrating a noticeably improved accuracy. In [4], The authors' primary focus was on predicting agricultural loan delinquency, examining the associated lending risk using machine learning techniques. They implemented various ML classification algorithms, including Random Forest, Logistic Regression, and Gaussian Naïve Bayes. The Random Forest algorithm yielded the highest accuracy of 87.2%, while Gaussian Naïve Bayes produced the best recall result of 80.6% in predicting individual loan risk percentages. In [5], The authors' main focus was on modeling car loan prepayment and assessing the risk involved in loan approval using supervised machine learning techniques. They utilized a machine learning classification algorithm to forecast the risk percentage for individual loan applicants. Employing ML algorithms such as Logistic Regression, they achieved an accuracy of 85%. The Logistic Regression model also demonstrated a recall result of 84% and a precision of 83% in classifying loan risks. In [6], The authors evaluated the loan approval risk using the Logistic Regression algorithm to forecast the risk percentage linked to loan provision for individuals, achieving an accuracy rate of 77%. In [7], The authors' primary focus was on predicting bank loan approvals using machine learning techniques. They utilized eight models: Logistic Regression, Gaussian NB, Random Forest, Decision Tree, SVM, K Neighbors Classifier, Gradient Boosting, and XGB Classifier. Common evaluation metrics for ML algorithms include accuracy, recall, and precision. The accuracy percentages for the ML models were: Logistic Regression (69.6%), Gaussian NB (44.3%), Random Forest (79.6%), Decision Tree (72.8%), SVM (77%), K Neighbors Classifier (74.9%), Gradient Boosting (81.1%), and XGB Classifier (80.6%). The Gradient Boosting technique particularly excelled, demonstrating a significantly improved accuracy of 81.1%. In [8], The authors employed seven models: Logistic Regression, Random Forest, Support Vector Machines, Bagging Classifier, LGBM, XGBoost, and LSTM. Imbalance Accuracy, recall, precision, and F1 Score were common metrics used in ML algorithms to assess model performance. The imbalance accuracies for the ML models were: Logistic Regression (−59%), Random Forest (45%), Support Vector Machines (−37%), Bagging Classifier (48%), LGBM (60%), XGBoost (58%), and LSTM Classifier (43%). In [9], The authors introduced a method for predicting loans using machine learning, employing the Logistic Regression algorithm with stratified k-folds cross-validation and Random Forest to estimate the risk percentage linked to loan provision for individuals. They achieved an accuracy of 72.1%, with a Logistic Regression F1 score of 82.8% and a Random Forest accuracy of 79.5%. In [10],The authors present a thorough comparative analysis of traditional machine learning techniques used to predict loan defaults. Their findings underscore the effectiveness of different algorithms in improving predictive accuracy, an essential factor for financial institutions. The study examines several machine learning models, including Random Forest and XGBoost, revealing that ensemble methods considerably surpass traditional credit scoring techniques in predicting loan defaults.

In [11], The authors present a deep learning model designed to assess loan repayment risk, demonstrating significant advancements in predictive accuracy over traditional approaches. Their research highlights the model's effectiveness in identifying high-risk borrowers, which is critical for financial institutions. This deep learning method outperformed conventional models, such as logistic regression, in predicting loan defaults. Utilizing a large dataset, the model improves its ability to generalize across diverse borrower profiles and economic situations. The results suggest that adopting deep learning techniques could greatly reduce default rates, thereby enhancing the overall stability of lending portfolios. In [12], The authors investigate the use of ensemble learning techniques in financial risk assessment, showcasing substantial enhancements in predictive accuracy. The study highlights the effectiveness of integrating multiple machine learning models to improve risk prediction. The ensemble model achieved a significant boost in accuracy, surpassing traditional single-model methods.In [13], The authors perform a comparative analysis of machine learning and deep learning models for credit risk assessment, revealing valuable insights into their performance. The study highlights that deep learning models demonstrate greater effectiveness than traditional machine learning methods in accurately predicting credit risk. Across multiple metrics, such as accuracy and AUC (Area Under Curve) scores, deep learning models consistently surpassed their machine learning counterparts. Additionally, the research employed large datasets, emphasizing the importance of data quality and preprocessing in achieving optimal model performance.

Previous studies relied solely on ML techniques without incorporating other AI methods and did not utilize all modern data preprocessing techniques. In this study, advanced data preprocessing methods are applied, and a range of AI techniques are utilized, including both ML and deep learning (DL) models.

To highlight the unique aspects and innovations of our approach compared to existing studies, we will explicitly outline the distinctive features of our methodology in our paper. These features include:

1. **Comprehensive Algorithmic Framework:** Unlike many existing studies that focus solely on either traditional Machine Learning or Deep Learning techniques, our study combines an extensive range of both ML and DL algorithms, providing a holistic approach to loan default prediction.
2. **Integration of Ensemble Methods:** We utilize a variety of individual classifiers and enhance prediction accuracy through Ensemble Methods. This innovative integration leverages the strengths of multiple algorithms, resulting in a more robust prediction framework.
3. **Advanced Data Balancing Techniques:** Our implementation of SMOTE and SMOTE-TOMEK techniques addresses class imbalance—an often-overlooked issue in loan default prediction studies. By ensuring our dataset is well-balanced, we improve the generalizability and reliability of our predictive models.
4. **Focused Evaluation Metrics:** We employ a comprehensive evaluation strategy that includes accuracy, precision, recall, and F1-measure, allowing for a nuanced assessment of model performance. This multifaceted approach distinguishes our research from others that may rely on a limited set of metrics.
5. **Empirical Findings of Model Performance:** Our empirical results highlight DenseNet and ResNet as the most accurate models, providing new insights into the effectiveness of these architectures in a financial context.

We believe that emphasizing these innovative aspects will clarify how our approach differs from and contributes to the field. We look forward to incorporating these elements into our paper.

III. METHODOLOGY

The financial sector is currently experiencing a flourishing integration of Artificial Intelligence applications. Institutions are harnessing the immense potential of AI to deliver business solutions and streamline processes, resulting in improved efficiency and elevated customer experiences. It is now evident that AI has become a crucial asset, providing a competitive edge through its sophisticated decision-making capabilities. As AI and ML continue to achieve remarkable milestones, it is clear that they stand on the brink of revolutionizing the banking industry [14]. The method put forward for forecasting banking customer loans consists of two main Phases, illustrated in Figure 1. The first phase encompasses data explanation, analysis, and preprocessing to ready it for the application of AI techniques. The second phase integrates AI techniques such as ML models, DL models, and ensemble learning to forecast banking customer loans using both selected features and the complete feature set. The proposed system comprises a series of sequential steps: Data Analysis, Data Preprocessing, Model Optimization, Application of Artificial Intelligence Models, and Model Evaluation.

Data has been collected from Kaggle, a leading source of data for educational purposes [15]. Training a model with a dataset involves splitting the dataset into two parts, typically in ratios like 80:20 or 85:15. The larger portion is used for training the model, while the smaller portion is kept for testing. The model's accuracy is then assessed based on this testing phase. The dataset is divided into an 80% training set and a 20% testing set to ensure the model's performance is evaluated on unseen data. This evaluation helps determine the model's ability to predict new, previously unseen data and assess if it has learned meaningful patterns from the training data. Random splitting is crucial to ensure both subsets represent the data's distribution and prevent biases in evaluation. While the 80-20 ratio is commonly used, it can be adjusted based on dataset size and specific AI requirements. In this study, the bank dataset comprises 252,000 records and 13 features, detailed in Table 1.

A. DATA ANALYSIS

Data analysis entails processing, encoding, categorizing, and organizing collected data to ensure its reliability, readiness, and suitability for thorough analysis. It includes computations and calculations based on specific parameters, examining the relationships within the data sets, and assessing study hypotheses using statistical tools, taking into account the patterns and relationships within the data [16]. A question arises about the criteria for deciding whether to approve or reject a loan application. Two target variables guide the loan approval process for customers. Before granting a loan, all necessary formalities, such as income proof, address proof, ID verification, and more, must be thoroughly reviewed to determine the customer's loan repayment eligibility. Loans are often vital for middle-class individuals, whether for education or business purposes. However, there are situations where individuals encounter unexpected financial crises, or some may attempt to deceive banks [17]. Therefore, it is essential to examine all these aspects thoroughly. The probability of loan repayment is higher if the customer profile is stronger, and background verification is crucial in ensuring timely loan repayment. Hence, an analysis is conducted based on several

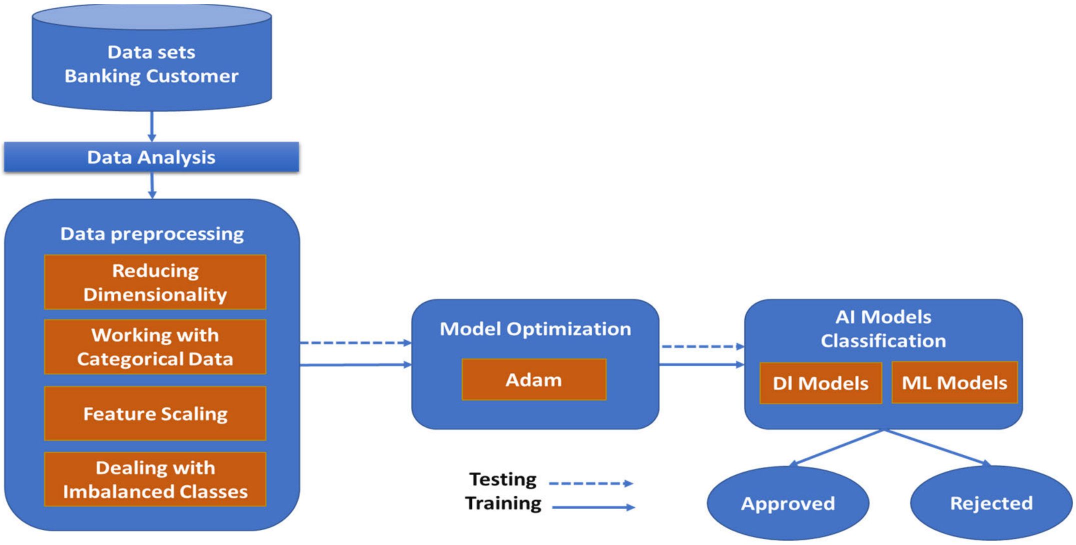

FIGURE 1. The proposed architecture.

TABLE 1. Description of the variables / features in the dataset.

No	Variables / Features	Description	Data Type
1	Id	Numbering the items in the dataset	İnteger
2	Income	User's income	İnteger
3	Age	User's age	İnteger
4	Experience	The number of years the user has in professional experience.	İnteger
5	Married/Single	The marital status is either " Single " or " Married".	String
6	House Ownership	Owned, rented, or neither.	String
7	Car Ownership	Is the person a car owner?	String
8	Profession	The type of work each individual engages in.	String
9	City	The city where the individual resides.	String
10	State	The state where the individual resides.	String
11	Current Job Years	The duration of experience in the current job.	İnteger
12	Current House Years	The length of time an individual has lived in their current residence.	İnteger
13	Risk Flag	Defaulted on a loan.	İnteger

factors that serve as the target variables in the loan approval process [18]. This paper utilizes a variety of graphics to visually depict the data analysis process, aiding in the understanding and exploration of the data.

1) KEY ATTRIBUTES

Figure 2. features a heatmap, showcasing both positive and negative attribute values. Heatmaps assist in analyzing interdependent attributes within the data. A heatmap visually represents data using colors to indicate values in a matrix or table. This graphical tool is valuable in data analysis, offering insights into patterns, relationships, and trends present in the data [19]. Heatmaps are commonly used for visualizing numerical data, requiring a matrix or table with rows and columns representing variables or categories, and cells containing the values to be displayed. A suitable color palette should be chosen, with warmer colors like red and yellow denoting higher values and cooler colors like blue and green indicating lower values. Proper labeling of rows and columns is essential for context. Heatmaps are effective in identifying patterns and trends in large datasets, offering a visual summary that aids in data exploration and hypothesis generation. They simplify complex data structures, facilitating communication of findings to non-technical audiences [20]. Heatmaps play a vital role in detecting trends and patterns, particularly in finance, to support data-driven decision-making.

2) DISTRIBUTION OF ATTRIBUTES

This pertains to the distribution of attribute or variable values within a dataset, offering insights into the range, patterns, and frequency of values for each attribute [21]. Understanding attribute distribution assists analysts in detecting trends, anomalies, and relationships among variables, which aids in decision-making and drawing meaningful conclusions from the data. Here's an example of this. a) Figure 3. illustrates the distribution of house ownership status based on the duration of current job years, exploring the correlation between house ownership status and the length of time in current employment. The aim of this analysis is to understand the distribution

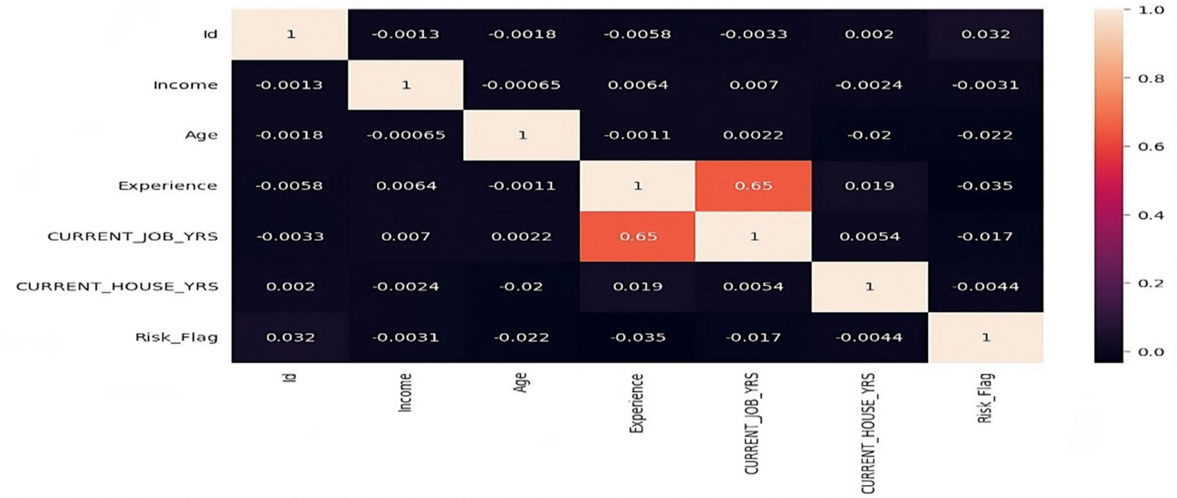

FIGURE 2. Heat map.

of house ownership status among individuals relative to their current job tenure.

b) Figure 4. showcases the distribution of current employment tenure among different categories of house ownership in the dataset. This analysis aims to examine how the duration individuals have spent in their current jobs is distributed across various house ownership statuses.

This chart is similar to the one in Fig. 3, but with a revised horizontal axis for better clarity. It illustrates the relationship between the duration of professional experience, ranging from one to a single year, and home ownership status (owned, rented, or neither). Thus, the chart helps clarify the connection between these variables, making the dataset easier to understand.

c) Figure 5 depicts a visual representation showcasing the distribution pattern of the target variable. This target variable holds importance as it defines how the values of the main variable being studied are dispersed within the dataset. Through a detailed analysis of this distribution, valuable insights are gained into the frequencies and patterns displayed by these values or categories. This analytical method enhances comprehension of the composition and characteristics linked to the target variable, revealing its attributes and aiding in the detection of significant trends or anomalies within the dataset. This step is pivotal in the data analysis process, enabling informed decision-making and the derivation of meaningful conclusions from the data.

B. DATA PREPROCESSING

The dataset exhibits data-related issues like outliers and noise, requiring a preprocessing stage to clean and enhance its quality. Perform data cleaning to remove any anomalies or outliers that could adversely affect the predictive models [22]. Numerous techniques are employed to enhance data quality, encompassing the management of categorical variables and the standardization of numerical features. These techniques involve dimensionality reduction, handling categorical data (including ordinal features and the 'get dummies' method), feature scaling, and addressing imbalanced classes. These techniques can be summarized briefly as follows:

1) REDUCING DIMENSIONALITY

This includes methods to reduce the dataset's feature count while retaining maximum information [23]. The influence of dimensionality reduction on various classification algorithms is examined using financial loan data [24]. Upon examining our dataset, we determined that the (Id) column is unnecessary. Therefore, we will proceed to remove it.

2) WORKING WITH CATEGORICAL DATA

It represents categorical data using binary vectors. In this method, each category is depicted as a binary vector with a length equal to the total number of categories [25]. Categorical data comprises non-numeric variables that denote categories or groups. When working with categorical data, two primary tasks are:

a: WORKING WITH ORDINAL FEATURES

Ordinal features are features defined by values derived from a categorical label set where an order relation is defined [26]. Ordinal features exhibit a natural order or ranking, and specialized methods are employed to manage them appropriately, preserving their ordinal relationships during analysis.

In loan prediction, effectively handling ordinal features involves encoding categories with a meaningful order using methods such as ordinal encoding, which assigns numerical values (e.g., assigning 1 to "Poor" and 4 to "Excellent" for credit ratings) to preserve the ranking, and applying advanced techniques like polynomial encoding to capture non-linear relationships or embeddings in deep learning to learn abstract patterns, thereby enhancing the model's ability to make accurate predictions and gain insights into borrower

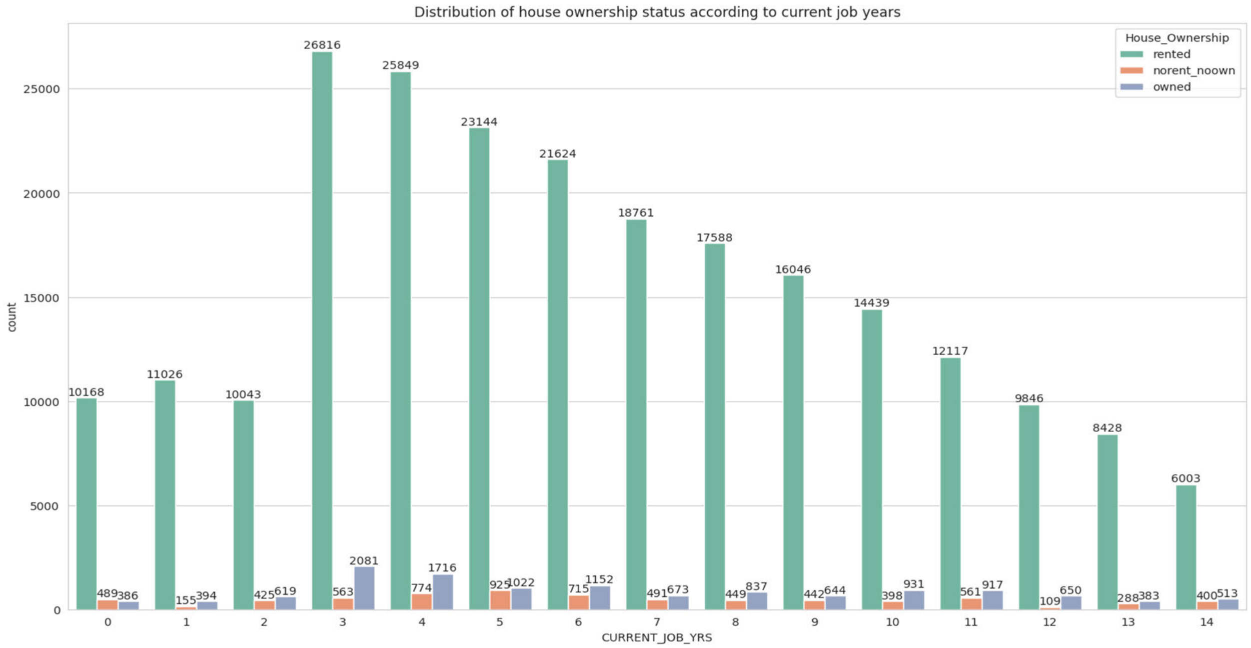

FIGURE 3. **Distribution of house ownership status according to current job years.**

behavior by properly reflecting the ordinal nature of the features.

b: 'GET DUMMIES' PROCESS

Also known as one-hot encoding, One-hot encoding uses a sparse vector, where only one element is set to 1, while the rest are set to 0. This method is often used to represent strings with a finite set of values. However, when applying one-hot encoding, high cardinality leads to feature vectors with high dimensions [27]. It is a method to transform categorical variables into binary representations, generating distinct binary columns for each category. This enables algorithms to effectively process categorical data.

The 'Get Dummies' process, also known as one-hot encoding, transforms categorical variables into binary vectors where each category is represented by a distinct binary column, with a 1 in the column corresponding to the category and 0s elsewhere, enabling algorithms to process categorical data effectively; for example, encoding a ''Loan Purpose'' feature with categories like ''Home,'' ''Car,'' ''Education,'' and ''Personal'' results in separate binary columns for each category, allowing models to differentiate and utilize these features in predictions. However, high cardinality can lead to high-dimensional feature vectors, such as when encoding a feature with many unique values like ''Loan Type,'' which can increase computational costs and model training time. To mitigate these issues, techniques like feature hashing can be used to reduce dimensionality, or embedding layers in deep learning can represent high-cardinality features as dense vectors, managing dimensionality while capturing complex relationships.

3) FEATURE SCALING

Feature scaling is an indispensable preprocessing step before model construction. Utilizing feature scaling techniques is vital for addressing challenges, as they involve adjusting values to enable straightforward and fair comparisons among them [28]. Feature scaling seeks to bring all numerical features to a comparable scale, preventing any single feature from overshadowing others. Common scaling techniques include Standardization, which scales features to have a mean of 0 and a standard deviation of 1.

Feature scaling is a crucial preprocessing step that adjusts numerical feature values to a comparable scale, ensuring that no single feature disproportionately influences the model. Standardization, which scales features to have a mean of 0 and a standard deviation of 1, is vital for algorithms like k-NN that are sensitive to feature scales, improving convergence and interpretability by providing balanced feature importance. Min-Max scaling, which maps features to a fixed range, usually [0, 1], is beneficial for neural networks to ensure equal contribution of features to gradient calculations and model stability. Robust scaling, which adjusts based on median and interquartile range, mitigates the influence of outliers and skewed distributions, making it suitable for features with extreme values. In machine learning and deep learning for loan prediction, these scaling techniques enhance model performance and convergence by treating features fairly and improving the accuracy of predictions.

4) DEALING WITH IMBALANCED CLASSES

Handling imbalanced datasets presents a substantial challenge. In imbalanced data, there is an uneven distribution

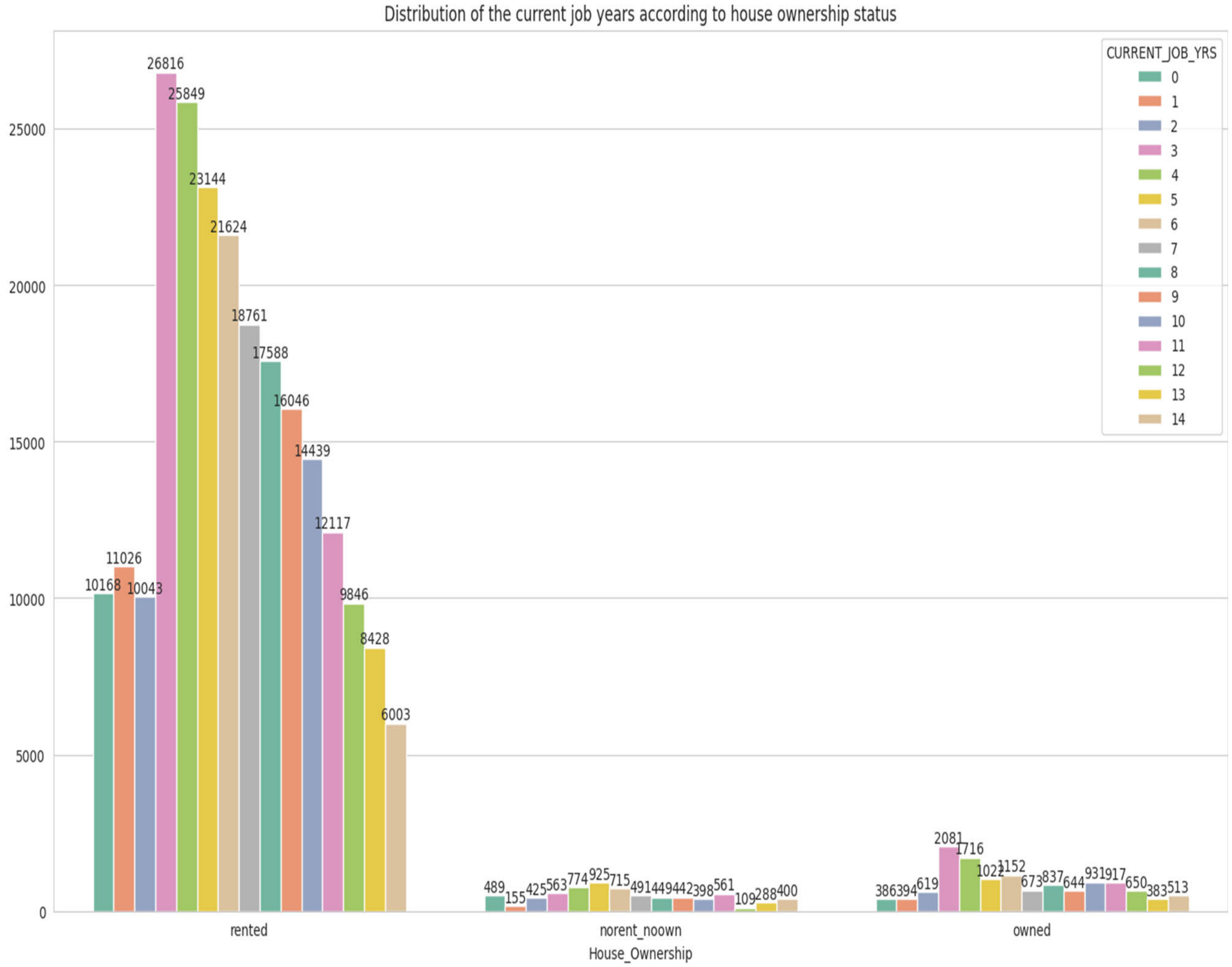

FIGURE 4. Distribution of the current job years according to house ownership status.

of samples across diverse classes, with the minority class having significantly fewer samples than the majority class [29]. In datasets with imbalanced class distribution, one class may considerably exceed the others, leading to biased models that perform poorly on underrepresented classes. To address this, oversampling (e.g., SMOTE) and undersampling techniques are employed to balance class proportions and enhance model performance. SMOTETomek combines the advantages of both SMOTE and Tomek links. In SMOTE, synthetic data is produced through a linear combination of two similar samples from the minority class, involving the random selection of one sample from the nearest neighbors of another within the minority class. The creation of these samples is based on operations in feature space, not data space [30]. Undersampling techniques allow for the evaluation of classifiers using smaller, representative subsets, yielding high-confidence metrics in reduced processing time. When classifiers struggle with large amounts of data from the entire dataset, undersampling offers a solution with significant variations in precision and recall. By utilizing undersampling techniques, it is feasible to assess only a portion of the data during the evaluation process, ensuring the model's representativeness and generalization are preserved [31]. Implementing these techniques improves the dataset's quality, addresses dimensionality challenges, manages categorical data efficiently, and establishes a stronger foundation for constructing predictive models and performing data analysis.

Dealing with imbalanced classes is a critical preprocessing step in loan prediction tasks, where datasets often exhibit uneven class distributions, with one class, such as loan defaults, being significantly underrepresented compared to the majority class, such as loan approvals. This imbalance can lead to biased models that perform poorly on the minority class. To address this, techniques like oversampling and undersampling are employed to balance class proportions. For instance, the Synthetic Minority Over-sampling Technique (SMOTE) generates synthetic samples by interpolating

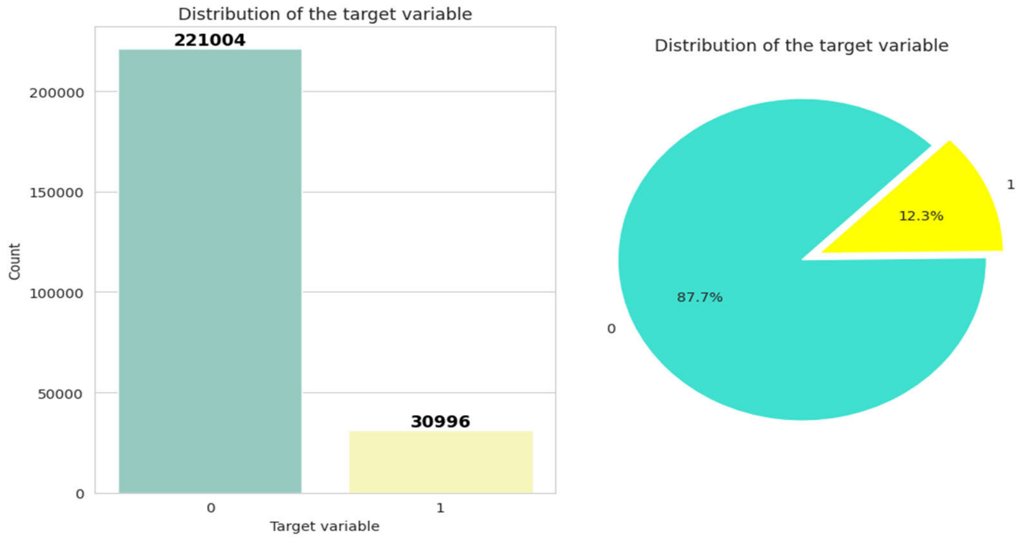

FIGURE 5. Distribution of the target variable (Risk Flag).

between existing minority class instances, creating new examples in the feature space by selecting one sample and generating synthetic points along the line segments connecting it to its nearest neighbors within the minority class. SMOTE-Tomek combines SMOTE with Tomek links, which identifies and removes borderline examples that are misclassified by other classes, further refining the dataset. Conversely, undersampling techniques reduce the number of majority class samples to create a more balanced dataset, allowing classifiers to be trained and evaluated on a representative subset of the data, thereby improving precision and recall without overwhelming computational resources. For example, if a dataset has 90% loan approvals and 10% loan defaults, applying SMOTE might generate synthetic loan defaults to balance the ratio, while undersampling might involve reducing the number of loan approvals to match the number of loan defaults. Implementing these techniques enhances model performance by ensuring that the classifier is trained on a balanced dataset, which improves its ability to generalize to both classes and provides more reliable metrics in terms of precision and recall.

C. MODEL OPTIMIZATION

This process aims to boost the performance of AI models by identifying the optimal parameter set that minimizes the error or loss function. This is crucial for enhancing the model's ability to predict accurately and generalize to new, unseen data. Various optimization algorithms are utilized to fine-tune these parameters during the training process. Below is a brief overview of the most effective optimization algorithms that consistently deliver superior performance. One notable algorithm is Adam (Adaptive Moment Estimation). AI plays a crucial role in addressing decision-making challenges. While Machine Learning and Deep Learning do not directly make decisions, they can be integrated to improve the optimization of more complex and intricate combinatorial models [32]. The Adam optimizer is a well-known algorithm that dynamically adjusts learning rates by utilizing adaptive gradient-based momentum updates that consider past gradients. In essence, this algorithm merges aspects of RMSProp and Stochastic Gradient Descent with Momentum (SGDM), using squared gradients to adjust the learning rate akin to RMSProp [33].

Model optimization is a critical process aimed at enhancing the performance of AI models by meticulously identifying the optimal parameter set that minimizes the error or loss function, which is essential for improving both accuracy and the model's ability to generalize to new, unseen data. This process involves several detailed steps and techniques. Initially, parameter tuning is performed using methods such as grid search, which exhaustively evaluates a predefined set of hyperparameter values to identify the best-performing combination based on validation performance; random search, which samples hyperparameter values randomly from specified ranges and often finds good results more efficiently than grid search; Bayesian optimization, which leverages probabilistic models to guide the search for optimal hyperparameters by focusing on promising regions of the parameter space; and Hyperband, which adaptively allocates resources to promising configurations by balancing exploration of new configurations with exploitation of known good ones. Additionally, optimization algorithms play a significant role in fine-tuning model parameters during training. Stochastic Gradient Descent (SGD) updates parameters by computing gradients from random subsets of the training data, though it can be slow and sensitive to learning rates. Momentum improves upon SGD by incorporating a fraction of the previous parameter update into the current update, accelerating convergence and smoothing the optimization trajectory. RMSProp addresses issues with varying gradient magnitudes by adjusting learning rates based on a moving average of squared gradients, which helps manage the learning rate for each parameter individually. The Adam (Adaptive Moment Estimation) optimizer combines the strengths of both RMSProp and SGD with momentum, dynamically adjusting learning rates based on the moving averages of both gradients and squared gradients. It initializes parameters, computes gradients, updates the first and second moment estimates (mean and uncentered variance), applies bias correction to these estimates, and finally adjusts model parameters using these corrected moments. This adaptive approach enables Adam to handle varying gradient magnitudes and learning rates efficiently, making it particularly effective for training deep neural networks and complex models. The integration of these optimization techniques enhances the model's ability to tackle decision-making challenges in Machine Learning and Deep Learning, ultimately leading to improved training efficiency, performance, and generalization capabilities.

D. APPLYING ARTIFICIAL INTELLIGENCE MODELS

In this study, the employed artificial intelligence models can be categorized into two techniques: ML models and DL models, as shown in Figure 6.

1) MACHINE LEARNING MODELS

ML classification techniques are a subset of supervised learning algorithms that predict categorical outcomes or labels based on input data. The primary goal of classification is to assign new instances to predefined classes or categories. Evaluating the predictive performance of a ML model during validation aims to ensure an accurate assessment of the model's real-world effectiveness when deployed [34]. Below is a concise overview of several common classification techniques:

a: GAUSSIAN NAIVE BAYES

It is a classification algorithm rooted in Bayes' theorem, employing probabilistic principles and assuming feature independence. This algorithm follows a Gaussian (normal) distribution, calculating the probability of each class based on the input features to predict the class with the highest probability [4]. The Gaussian Naive Bayes is a variant of the Naive Bayes algorithm that integrates the Gaussian normal distribution, allowing it to manage continuous data within the realm of classification [35].

Gaussian Naive Bayes (Gaussian NB) Rationale: Gaussian NB is a probabilistic classifier based on Bayes' theorem, assuming that the features follow a Gaussian distribution. It is particularly effective for datasets with continuous features, making it suitable for our dataset, which includes numerical attributes like age, income, and credit score.

Characteristics:
- Fast and efficient for large datasets.
- Performs well with high-dimensional data.
- Robust to irrelevant features.

b: AdaBoost (ADAPTIVE BOOSTING)

Adaboost is a form of ensemble learning machinery that utilizes classifiers comprising a variety of base models [36].It is an ensemble learning algorithm that constructs a strong classifier by combining multiple weak classifiers. It iteratively trains weak classifiers on different subsets of the training data, giving higher weight to misclassified instances in each iteration. The final prediction is derived from aggregating the predictions of all weak classifiers [3].

c: GRADIENT BOOSTING

It is an ensemble learning algorithm that amalgamates multiple weak prediction models, often decision trees, to create a strong predictive model. This is accomplished by sequentially training weak models, with each subsequent model correcting the errors made by its predecessor [37]. The final prediction is derived from the collective predictions of all weak models [8].

Gradient Boosting Rationale: Gradient Boosting is an ensemble technique that builds models sequentially, focusing on correcting the errors of previous models. It is effective for handling various types of data and can improve predictive performance significantly.

Characteristics:
- High accuracy and robustness against overfitting.
- Can handle mixed data types (numerical and categorical).
- Provides feature importance, aiding in interpretability.

d: K NEIGHBORS CLASSIFIER

K Nearest Neighbors is advantageous in pattern recognition assessments as it classifies a data point by examining the classifications of its neighbors and retaining all available cases. Although primarily suited for classification tasks, it can also be utilized in regression scenarios [38]. It is a non-parametric algorithm that categorizes new instances based on their resemblance to the training data. By examining the class labels of the k nearest neighbors in the feature space, it assigns a class label to a new instance. The classification outcomes are swayed by the user-defined parameter k [2], [15].

This model excelled due to its ability to leverage local information, effectively capturing relationships among similar applicants. However, it may struggle with high-dimensional data, which can lead to overfitting.

e: LOGISTIC REGRESSION

Logistic regression is a statistical analysis method that constructs a statistical model to clarify the relationship between a binary or dichotomous outcome (such as yes/no) – the dependent or response variable – and a set of independent predictor or explanatory variables [39]. It is an algorithm employed in supervised learning for binary classification tasks, modeling the relationship between the independent variables and the probability of the binary outcome using the logistic function. The algorithm estimates the coefficients of the independent

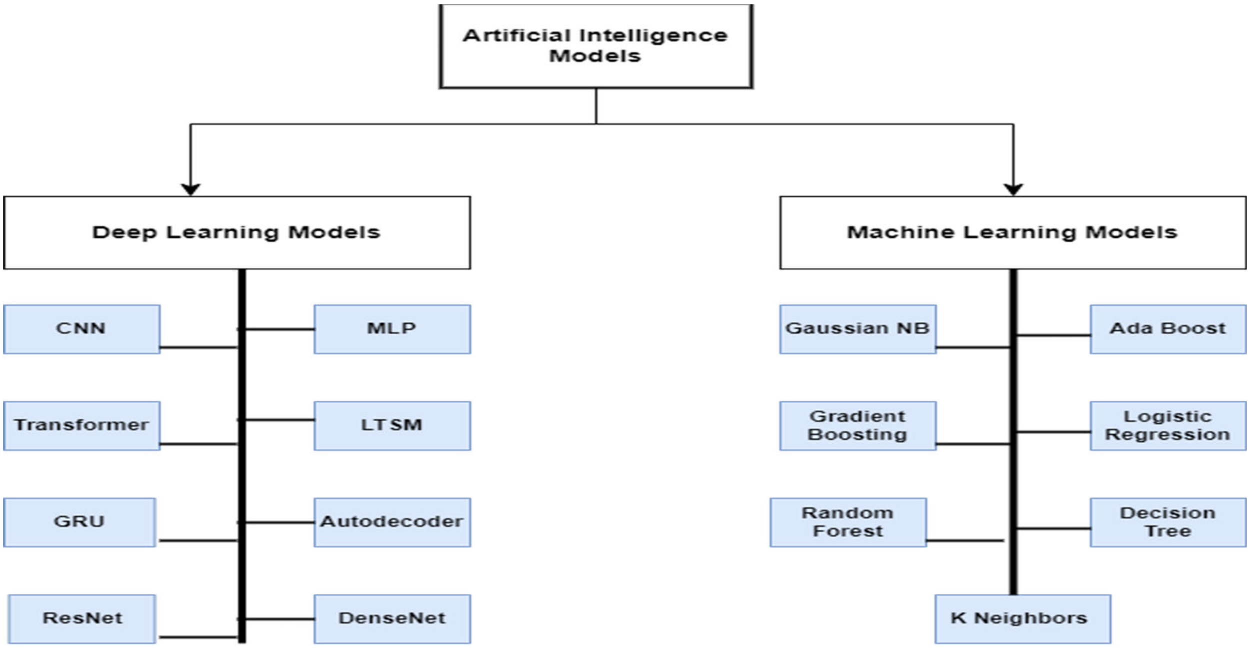

FIGURE 6. **Illustration of the Utilized AI models.**

variables through maximum likelihood estimation and predicts the class based on a threshold probability [2], [5].

Logistic Regression Rationale: Logistic Regression is a widely used statistical method for binary classification problems. It estimates the probability of a binary outcome based on one or more predictor variables. Given the binary nature of the loan default status (default/no default), this algorithm is a natural choice.

Characteristics:

- Interpretable coefficients, allowing for easy understanding of feature impacts.
- Works well with linearly separable data.
- Less prone to overfitting with a small number of features.

f: DECISION TREES

The fundamental algorithmic principle of a call tree dictates that all attributes or options must be evaluated. Feature selection is based on maximizing the information gain from these options. The data in a call tree is organized as IF-THEN rules. This model is an extension of the C4.5 classification algorithm developed by Quinlan [2].

g: RANDOM FOREST

Random forests are a classification learning framework used for both classification and regression tasks. They work by constructing a large number of decision trees during training and then determining the final class by taking a majority vote from the predictions of these individual trees [2].

2) DEEP LEARNING MODELS

Deep learning (DL) has revolutionized the landscape of artificial intelligence (AI), offering solutions to enduring challenges in the AI domain. DL models, evolved forms of Artificial Neural Networks (ANNs) with multiple layers—be they linear or non-linear—have effectively addressed intricate problems. These interconnected layers, with diverse weights, establish connections between different levels, enabling DL models to learn hierarchical features from various data types. This capability positions DL models as powerful solutions for tasks like recognition, regression, semi-supervised, and unsupervised learning [40]. DL classification methods encompass a subset of DL algorithms tailored for addressing classification challenges. These techniques harness artificial neural networks, which are DL models inspired by the human brain's neural network structure. Below is a concise overview of some common DL classification techniques:

a: MLP (MULTI-LAYER PERCEPTRON)

It is a conventional feedforward neural network, incorporating one or more hidden layers, capable of addressing diverse classification tasks and learning intricate relationships within the data [41]. It is a conventional feedforward neural network, incorporating one or more hidden layers, capable of addressing diverse classification tasks and learning intricate relationships within the data [42].

Multi-Layer Perceptron (MLP) Rationale: MLP is a type of neural network that can capture complex relationships in the data. It is suitable for datasets with non-linear relationships, which is often the case in financial data.

Characteristics:

- Capable of learning intricate patterns through multiple layers.

- Flexible architecture that can be adjusted based on the complexity of the dataset.

- Effective for both classification and regression tasks.

b: CNN (CONVOLUTIONAL NEURAL NETWORK)

CNN, as one of the widely used deep learning networks, has significantly bolstered the current prominence of DL. Its ability to autonomously identify significant features without human intervention sets it apart from its predecessors, leading to extensive use. Therefore, we will explore CNN by explaining its fundamental components [43]. It is frequently utilized for classification tasks, using convolutional layers to automatically extract spatial features from input data, rendering it highly effective in various artificial intelligence (AI) applications [44].

Convolutional Neural Network (CNN) Rationale: Although CNNs are primarily used for image data, they can also be applied to structured data by treating it as a grid. In this study, CNNs can capture local patterns and interactions between features, which may be beneficial for understanding complex relationships in loan applications.

Characteristics:

- Excellent at capturing spatial hierarchies in data.

- Reduces the number of parameters through weight sharing, making it efficient.

- Robust to noise and variations in input data.

c: LSTM (LONG SHORT-TERM MEMORY)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network recognized for its ability to capture and utilize information from previous time steps for future predictions, particularly in forecasting scenarios with extensive historical data. The ‘‘memory line’’ within LSTM architecture enables the retention of prior stages, facilitated through gates with an extended memory line [45]. LSTM is characterized as a type of recurrent neural network that excels at processing sequential data, specifically designed to handle long-range dependencies in sequences, contributing to its popularity in tasks such as these [8].

d: TRANSFORMER

The transformer, a neural network primarily constructed on self-attention mechanisms, enables the creation of relationships between distinct features [46]. It is a robust architecture for handling sequential data, leveraging self-attention mechanisms to capture long-range dependencies, resulting in cutting-edge performance across a range of artificial intelligence (AI) applications [47].

e: GRU (GATED RECURRENT UNIT)

The GRU model, a distinct variant of recurrent neural networks, addresses long-term dependence issues, forming the basis for an ensemble learning method designed for analyzing structure responses in the context of time-varying uncertainties [48]. Moreover, by utilizing GRU networks, an active learning strategy is introduced and applied to improve model accuracy while concurrently decreasing the necessary training data volume GRU is described as another type of Recurring Neural Network, simpler than LSTM but still effective in processing sequential data, characterized by fewer parameters and reduced computational cost [49].

f: AUTOENCODER

An unsupervised learning model, capable of independently identifying data features from extensive sample sets, serves as a dimensionality reduction method [50]. Within the domain of unsupervised learning, models like the autoencoder depend solely on input training data. The evolution of deep learning has generated significant scholarly interest in the autoencoder, leading researchers to develop various improved versions customized for different application domains. Fundamentally, the autoencoder strives to learn representation functions for datasets, facilitating the creation of models that encapsulate the acquired insights from the data [50]. They reconstruct input data from a condensed representation, proving valuable for dimensionality reduction and feature extraction purposes [41].

g: ResNet (RESIDUAL NEURAL NETWORK)

ResNet, also known as Residual Neural Network, effectively tackles the vanishing gradient problem through the incorporation of skip connections, enabling the training of exceptionally deep networks. This feature contributes to its widespread use in various artificial intelligence (AI) tasks [51]. ResNet improves upon traditional neural networks by introducing ‘‘loops’’ via residual connections, creating a more complex model. Empirical evidence shows that these residual connections significantly reduce training errors, accelerate training, and maintain generalization capabilities [52].

h: DenseNet (DENSELY CONNECTED CONVOLUTIONAL NETWORK)

DenseNet is a convolutional neural network architecture that forges dense connections between every layer, promoting feature reuse and enhancing information flow, ultimately resulting in improved performance with fewer parameters [53]. DenseNet distinguishes itself as a problem-specific network architecture, utilizing a multi-level innovative fine-tuning strategy to create several specialized networks. Its core concept involves forming connections between each layer and all preceding layers, enhancing architectural flow. A notable advantage of DenseNet is its reduced parameter set compared to other deep learning models, simplifying complexity and increasing accuracy [54].

These deep learning (DL) techniques have significantly propelled the machine learning (ML) field, delivering remarkable outcomes across various classification challenges and domains. DL classification methods have transformed the ML landscape, demonstrating impressive performance in banking applications. They can autonomously learn complex patterns and representations from raw data, making them formidable tools for effectively addressing intricate classification problems across diverse domains.

DenseNet's superior performance can be attributed to its architecture, which promotes efficient feature reuse and addresses the vanishing gradient problem. This allows the model to learn complex patterns effectively, although it requires more computational resources and longer training times.

IV. RESULTS

This section explores the application of Artificial Intelligence Models to the dataset, encompassing the training and testing results of implementing ML algorithms like Gaussian NB, Logistic Regression, Ada Boost, Gradient Boosting, Decision Trees, Random Forest, and K Neighbors Classifier on all features. It also discusses the testing and training results of applying DL algorithms such as MLP, CNN, LSTM, Transformer, GRU, Autoencoder, ResNet, and DenseNet. The models' evaluation utilizes four standard metrics: Accuracy (AC), precision (PR), recall (RE), and F1-score (FS), as outlined in Equations 1-4. These metrics rely on the terms TP (true positive), TN (true negative), FP (false positive), and FN (false negative).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1 Score} = \frac{2.\,\text{precision}.\,\text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

$$\text{True positive rate (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

$$\text{False positive rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{6}$$

$$\text{Matthews correlation coefficient (MCC)} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})\,.\,(\text{TP} + \text{FN})\,.\,(\text{TN} + \text{FP})\,.(\text{TN} + \text{FN})}} \tag{7}$$

Accuracy is calculated by adding the true positive and true negative predictions (TP + TN) and dividing by the total dataset count (P + N). A perfect accuracy score is represented as 1, with the lowest achievable accuracy being 0 [55]. Precision is computed by dividing the count of accurate positive predictions (TP) by the total number of outcomes classified as positive (TP + FP). The highest achievable precision score is 1, with the lowest being 0 [55]. Recall is determined by the ratio of true positive predictions (TP) to the total count of positives (P), also known as Sensitivity. The optimal TP Rate is 1, with the minimum being 0 [55]. The F1 score, derived from the harmonic mean formula applied to precision and recall, spans between 0 and 1 [56]. The minimum value occurs when all positive samples are misclassified (TP = 0), and the maximum value is achieved with perfect classification (FN = FP = 0) [56].

The AUC-ROC provides a way to visualize a machine learning classifier's performance. The ROC curve helps filter out noise from the signal, while the AUC summarizes the ROC curve and reflects the classifier's ability to distinguish between distinct classes. As the AUC value increases, the model's effectiveness in differentiating between positive and negative classes improves [57].

TPR and FPR are plotted across various classification thresholds. As the classification threshold rises, more items are classified as positive, leading to an increase in both true positives and false positives. The area under the ROC curve (AUC) quantifies the space beneath the ROC curve [57].

The AUC, or Area Under the ROC Curve, measures a model's ability to distinguish between positive and negative classes. Its value ranges from 0 to 1:

An AUC of 1 indicates a perfect model that accurately separates the two classes. An AUC of 0.5 represents a model with no distinguishing power, similar to random guessing. An AUC below 0.5 implies a model that performs worse than random guessing [57].

Matthews Correlation Coefficient (MCC): The MCC provides a balanced metric that remains useful even with varying class sizes. A coefficient of +1.0 indicates a perfect prediction, 0.0 means the performance is no better than random chance, and −1.0 represents the worst possible prediction. It combines both accuracy and coverage into a single, balanced measure [58].

A. RESULTS OF ML EXPERIMENT ALGORITHMS

Seven classification algorithms, including K Neighbors Classifier, Logistic Regression, Gaussian NB, Gradient Boosting, Decision Trees, Random Forest, and Ada Boost, were employed as shown in Table 2, which presents the training and testing results for the dataset. The Random Forest exhibited the best performance in both training and testing. In terms of training results, the Random forest achieved the highest scores (AC = 100%, PR = 100%, RE = 100%, FS = 100%, AUC-Roc = 100%, MCC = 99%). the Decision Trees achieved the highest scores (AC = 100%, PR = 100%, RE = 100%, FS = 100%, AUC-Roc = 100%, MCC = 100%).the K Neighbors Classifier performed moderately well (AC = 95%, PR = 95%, RE = 95%, FS = 95%, AUC-Roc = 99%, MCC = 91%). Logistic Regression performed moderately well (AC = 84%, PR = 87%, RE = 84%, FS = 84%, AUC-Roc = 90%, MCC = 71%), while Gaussian NB achieved (AC = 81%, PR = 81%, RE = 81%, FS = 81%, AUC-Roc = 85%, MCC = 62%). Gradient Boosting reached (AC = 79%, PR = 79%, RE = 79%, FS = 79%, AUC-Roc = 86%, MCC = 58%), and Ada Boost displayed the lowest performance (AC = 77%, PR = 77%, RE = 77%, FS = 77%, AUC-Roc = 84%, MCC = 57%). In terms of testing results, the Random forest again achieved the highest scores (AC = 90%, PR = 89%, RE = 90%, FS = 90%, AUC-Roc = 91%, MCC = 48%). the Decision Trees yielded a

TABLE 2. The results of applying ML models.

Used Algorithms	Train Data						Test Data					
	Accuracy	Precision	Recall	F1-Score	AUC-Roc	MCC	Accuracy	Precision	Recall	F1-Score	AUC-Roc	MCC
Gaussian NB	0.81	0.81	0.81	0.81	0.85	0.62	0.76	0.80	0.76	0.78	0.60	0.09
Logistic Regression	0.84	0.87	0.84	0.84	0.90	0.71	0.86	0.80	0.86	0.82	0.62	0.07
Ada Boost	0.77	0.77	0.77	0.77	0.84	0.57	0.74	0.80	0.74	0.76	0.58	0.06
Gradient Boosting	0.79	0.79	0.79	0.79	0.86	0.58	0.75	0.81	0.75	0.77	0.63	0.1
K Neighbors Classifier	**0.95**	**0.95**	**0.95**	**0.95**	**0.99**	**0.91**	**0.88**	**0.89**	**0.88**	**0.89**	**0.87**	**0.48**
Decision Trees	1.0	1.0	1.0	1.0	1.0	1.0	0.87	0.89	0.88	0.88	0.74	0.47
random forest	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.99**	**0.90**	**0.89**	**0.90**	**0.90**	**0.91**	**0.48**

performance level of (AC = 87%, PR = 89%, RE = 88%, FS = 88%, AUC-Roc = 91%, MCC =48%). the K Neighbors Classifier again demonstrated the highest performance (AC = 88%, PR = 89%, RE = 88%, FS = 89%, AUC-Roc = 87%, MCC = 48%). Logistic Regression yielded a performance level of (AC = 86%, PR = 80%, RE = 86%, FS = 82%, AUC-Roc = 62%, MCC = 7%), Gaussian NB achieved (AC = 76%, PR = 80%, RE = 76%, FS = 78%, AUC-Roc = 60%, MCC = 9%), and Gradient Boosting reached (AC = 75%, PR = 81%, RE = 75%, FS = 77%, AUC-Roc = 63%, MCC = 1%). Ada Boost exhibited the lowest testing performance (AC = 74%, PR = 80%, RE = 74%, FS = 76%, AUC-Roc = 58%, MCC = 6%).

In Figure 7, a comparative analysis of the ML classifiers based on evaluation metrics is presented. The outcomes were promising, with the K Neighbors Classifier attaining the highest accuracy among all classifiers. While Logistic Regression, Gaussian NB, and Gradient Boosting exhibited comparable performance, they were not identical. In contrast, Ada Boost displayed lower accuracy.

B. RESULTS OF DL WITHOUT ANY BALANCED TECHNIQUES

The application of eight DL classification algorithms, including MLP, CNN, LSTM, GRU, CNN-LSTM, Bidirectional LSTM, ResNet, and DenseNet, to the dataset is presented in Table 3, with results shown for both training and testing in the absence of any balanced techniques.

This approach is not dependable as balanced data is essential. However, it was conducted to highlight the difference in results between using balanced data and not using it. DenseNet showcased the most effective performance in both training and testing. In terms of training results, DenseNet achieved the highest scores (AC = 91%, PR = 84%, RE = 71%, FS = 75%, AUC-Roc = 95%, MCC = 60%). ResNet performed at (AC = 91%, PR = 83%, RE = 72%, FS = 76%, AUC-Roc = 94%, MCC = 55%), MLP achieved (AC = 91%, PR = 80%, RE = 74%, FS = 76%, AUC-Roc =95%, MCC =53%), and CNN reached (AC = 89%, PR = 76%, RE = 70%, FS = 72%, AUC-Roc = 89%, MCC = 34%). Conversely, LSTM, GRU, CNN-LSTM, and Bidirectional LSTM showed the lowest performance (AC = 88%, PR = 44%, RE = 50%, FS = 47%, AUC-Roc =51%, MCC =0%).Regarding testing results, DenseNet again demonstrated the highest performance (AC = 91%, PR = 83%, RE = 71%, FS = 76%, AUC-Roc =94%, MCC =58%). ResNet achieved (AC = 90%, PR = 80%, RE = 70%, FS = 73%, AUC-Roc =93%, MCC =53%), MLP achieved (AC = 89%, PR = 76%, RE = 70%, FS = 72%, AUC-Roc = 91%, MCC =43%), and CNN reached (AC = 89%, PR = 74%, RE = 68%, FS = 70%, AUC-Roc = 87%, MCC = 31%). In contrast, LSTM, GRU, CNN-LSTM, and Bidirectional LSTM exhibited the same lowest performance (AC = 88%, PR = 44%, RE = 50%, FS = 47%, AUC-Roc = 51%, MCC = 0%). Figure 8 presents a comparative analysis of the DL classifiers based on evaluation metrics. The outcomes were satisfactory, with DenseNet and ResNet achieving the highest accuracy among all classifiers. MLP and CNN demonstrated similar performance, while LSTM, GRU, CNN-LSTM, and Bidirectional LSTM showed lower accuracy.

C. RESULTS OF DL WITH SMOTE TOMEK OF BALANCED TECHNIQUES

Table 4 presents the outcomes of applying DL with SMOTE TOMEK Balanced Techniques to the dataset for both training and testing. DenseNet and ResNet showcased the most effective performance in both training and testing. In terms of training results, DenseNet and ResNet achieved the highest scores (AC = 95%, PR = 95%, RE = 95%, FS = 95%, AUC-Roc = 99%, MCC = 91%). MLP and CNN achieved commendable scores (AC = 94%, PR = 94%, RE = 94%, FS = 94%, AUC-Roc = 99%, MCC = 89%). CNN-LSTM achieved a performance level of (AC = 91%, PR = 91%,

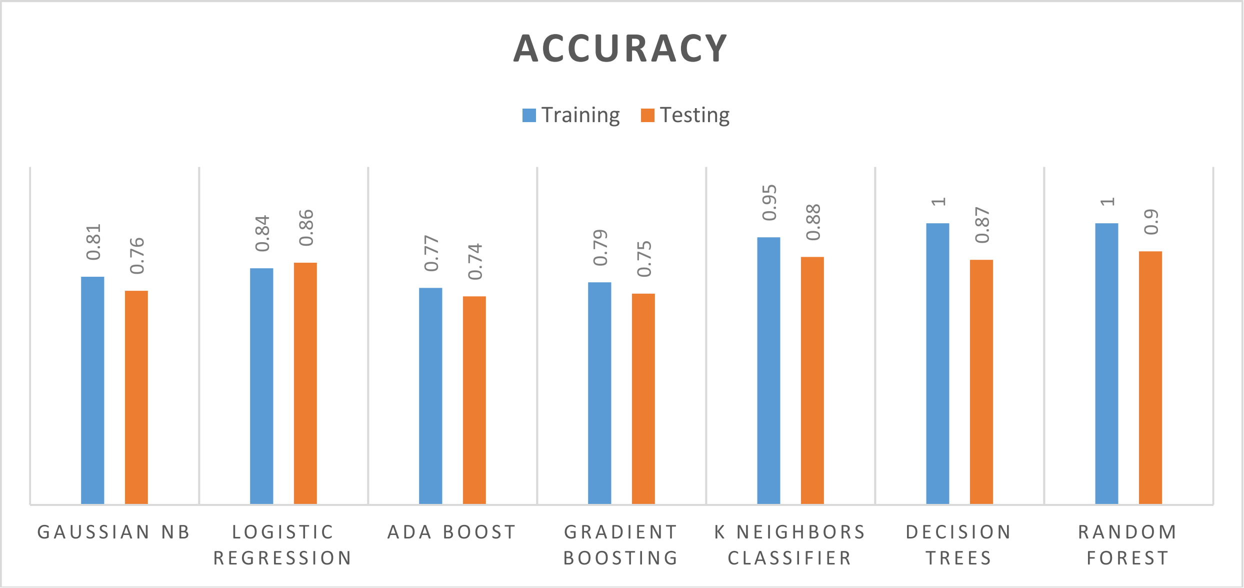

FIGURE 7. Chart comparing the performance of ML models.

TABLE 3. Results of DL without any balanced techniques.

Used Algorithms	Train Data						Test Data					
	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
MLP	0.91	0.80	0.74	0.76	0.95	0.53	0.89	0.76	0.70	0.72	0.91	0.43
CNN	0.89	0.76	0.70	0.72	0.89	0.34	0.89	0.74	0.68	0.70	0.87	0.31
LSTM	0.88	0.44	0.50	0.47	0.50	0.00	0.88	0.44	0.50	0.47	0.50	0.00
GRU	0.88	0.44	0.50	0.47	0.50	0.00	0.88	0.44	0.50	0.47	0.50	0.00
CNN-LSTM	0.88	0.44	0.50	0.47	0.51	0.00	0.88	0.44	0.50	0.47	0.51	0.00
Bidirectional LSTM	0.88	0.44	0.50	0.47	0.52	0.00	0.88	0.44	0.50	0.47	0.53	0.00
ResNet	0.91	0.83	0.72	0.76	0.94	0.55	0.90	0.80	0.70	0.73	0.93	0.53
DenseNet	**0.91**	**0.84**	**0.71**	**0.75**	**0.95**	**0.60**	**0.91**	**0.83**	**0.71**	**0.76**	**0.94**	**0.58**

RE = 91%, FS = 91%, AUC-Roc =85%, MCC = 71%). GRU achieved a performance level of (AC = 90%, PR = 91%, RE = 90%, FS = 90%, AUC-Roc =94%, MCC = 74%). Bidirectional LSTM attained a performance level of (AC = 85%, PR = 87%, RE = 85%, FS = 85%, AUC-Roc = 99%, MCC = 88%), while LSTM and Transformer exhibited the lowest performance. LSTM achieved (AC = 84%, PR = 88%, RE = 84%, FS = 83%, AUC-Roc = 92%, MCC =72%), and Transformer achieved (AC = 84%, PR = 86%, RE = 84%, FS = 84%, AUC-Roc =91%, MCC = 71%).

In terms of testing results, DenseNet and ResNet once again achieved the highest scores. DenseNet achieved (AC = 91%, PR = 79%, RE = 80%, FS = 79%, AUC-Roc = 94%, MCC = 58%), and ResNet achieved (AC = 91%, PR = 80%, RE = 77%, FS = 79%, AUC-Roc =95%, MCC = 60%). MLP and CNN also performed well. MLP achieved (AC = 89%, PR = 75%, RE = 80%, FS = 77%, AUC-Roc = 92%, MCC = 52%), and CNN achieved (AC = 89%, PR = 74%, RE = 80%, FS = 77%, AUC-Roc = 90%, MCC = 54%). CNN-LSTM performed at (AC = 88%, PR = 72%, RE = 67%, FS = 69%, AUC-Roc = 52%, MCC = −0.2%), GRU achieved (AC = 87%, PR = 70%, RE = 65%, FS = 67%, AUC-Roc = 72%, MCC = 18%), and Bidirectional LSTM achieved (AC = 87%, PR = 62%, RE = 53%, FS = 53%, AUC-Roc = 92%, MCC =54%). LSTM and Transformer displayed the lowest performance. Transformer achieved (AC = 86%, PR = 59%, RE = 52%, FS = 52%, AUC-Roc = 65%, MCC = 9%), and LSTM achieved (AC = 87%, PR = 48%, RE = 50%, FS = 47%, AUC-Roc = 66%, MCC = 11%). Nine DL classification algorithms were employed, including MLP, CNN, LSTM, GRU, Transformer,

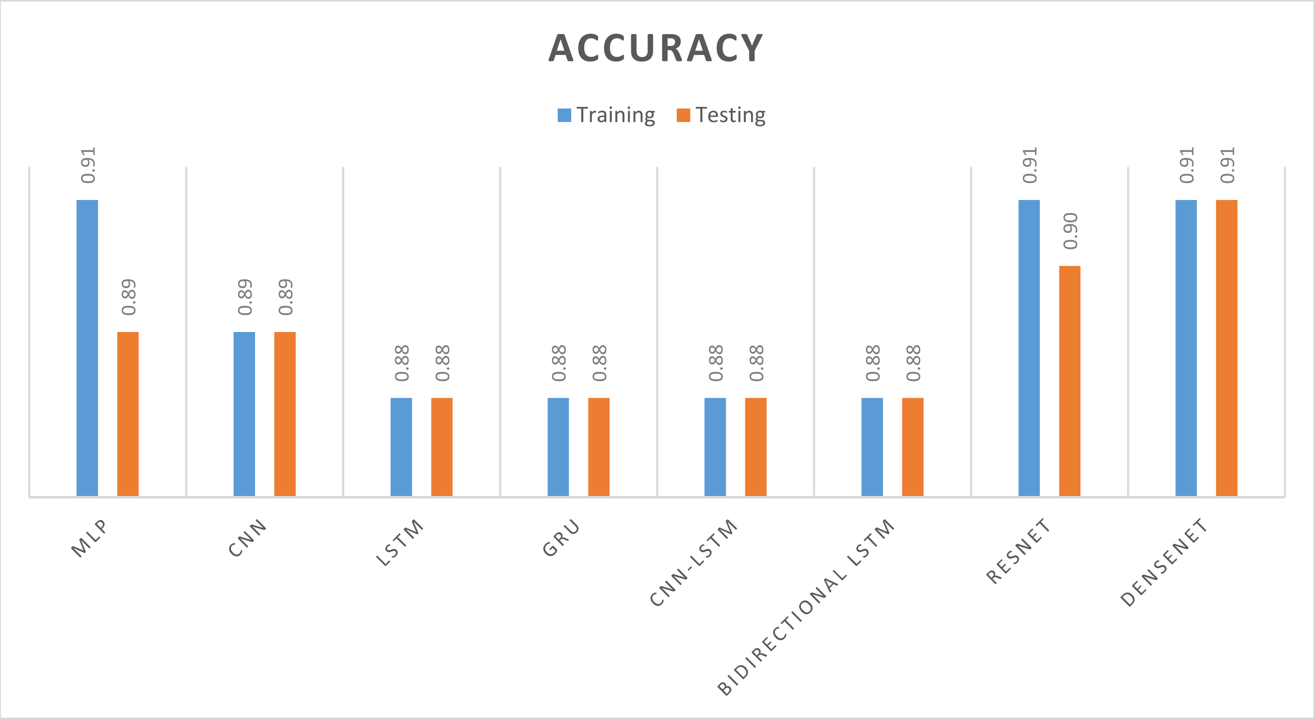

FIGURE 8. Performance comparison chart of DL without any balancing techniques.

TABLE 4. Results of DL with SMOTE TOMEK of balanced techniques.

Used Algorithms	Train Data						Test Data					
	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
MLP	0.94	0.94	0.94	0.94	0.99	0.89	0.89	0.75	0.80	0.77	0.92	0.52
CNN	0.94	0.94	0.94	0.94	0.99	0.88	0.89	0.74	0.80	0.77	0.90	0.54
LSTM	0.84	0.88	0.84	0.83	0.92	0.72	0.87	0.48	0.50	0.47	0.66	0.11
GRU	0.90	0.91	0.90	0.90	0.94	0.74	0.87	0.70	0.65	0.67	0.72	0.18
Transformer	0.84	0.86	0.84	0.84	0.91	0.71	0.86	0.59	0.52	0.52	0.65	0.09
CNN-LSTM	0.91	0.91	0.91	0.91	0.85	0.71	0.88	0.72	0.67	0.69	0.52	-0.002
Bidirectional LSTM	0.85	0.87	0.85	0.85	0.99	0.88	0.87	0.62	0.53	0.53	0.92	0.54
ResNet	0.95	0.95	0.95	0.95	0.99	0.91	0.91	0.80	0.77	0.79	0.95	0.60
DenseNet	0.95	0.95	0.95	0.95	0.99	0.90	0.91	0.79	0.80	0.79	0.94	0.58

CNN-LSTM, Bidirectional LSTM, ResNet, and DenseNet. Figure 9 illustrates the comparative analysis of DL classifiers based on evaluation metrics. The outcomes were satisfactory, with DenseNet and ResNet achieving the highest accuracy among all classifications. MLP and CNN demonstrated similar, good performance results. GRU and CNN-LSTM also had comparable performance results. LSTM, Transformer, and Bidirectional LSTM exhibited lower accuracy.

D. RESULTS OF DL WITH SMOTE OF BALANCED TECHNIQUES

Table 5 presents the outcomes of employing eight DL classification algorithms, namely MLP, CNN, LSTM, GRU, CNN-LSTM, Bidirectional LSTM, ResNet, and DenseNet, with SMOTE Balanced Techniques applied to the dataset for both training and testing.

DenseNet showed the most effective performance in both training and testing. In terms of training results, DenseNet achieved the highest scores (AC = 96%, PR = 96%, RE = 96%, FS = 96%, AUC-Roc = 99%, MCC =90%), followed by ResNet with scores of (AC = 95%, PR = 95%, RE = 95%, FS = 95%, AUC-Roc = 92%, MCC = 90%). MLP and CNN both achieved good scores (AC = 94%, PR = 94%, RE = 94%, FS = 94%, AUC-Roc = 99%, MCC = 89%). GRU performed at (AC = 90%, PR = 91%, RE = 90%, FS = 90%, AUC-Roc =94%, MCC = 76%), CNN-LSTM

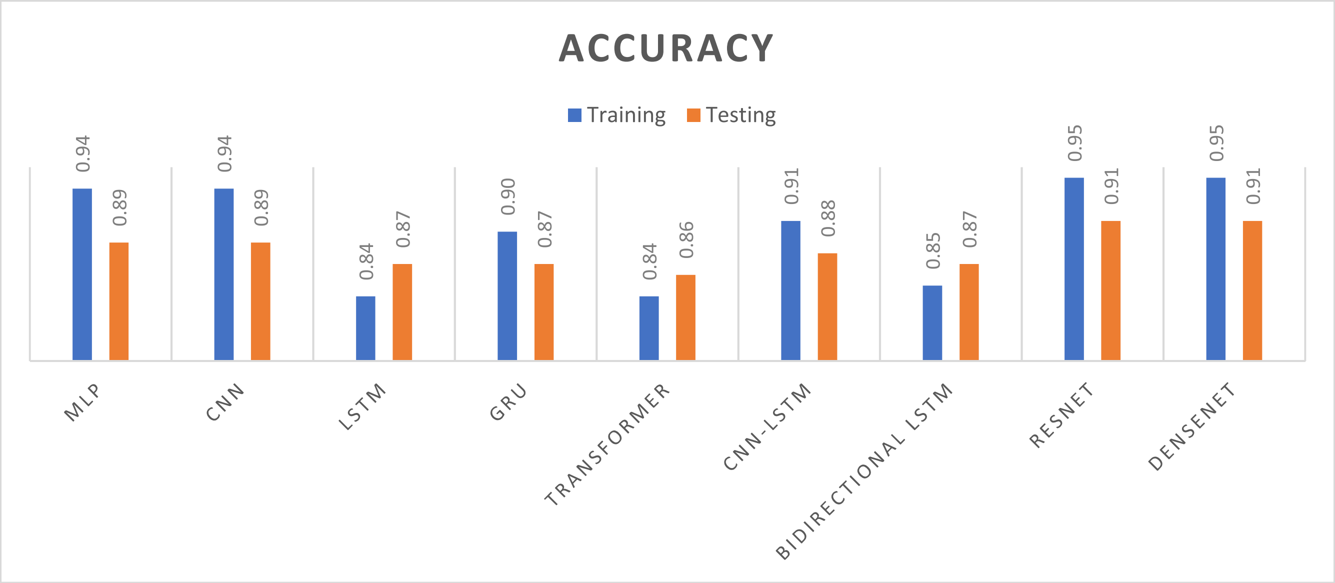

FIGURE 9. **Performance comparison chart of DL with SMOTE TOMEK balanced techniques.**

at (AC = 89%, PR = 89%, RE = 89%, FS = 89%, AUC-Roc =9 6%, MCC = 84%), and LSTM at (AC = 85%, PR = 86%, RE = 85%, FS = 84%, AUC-Roc =89%, MCC = 71%). Bidirectional LSTM exhibited the lowest performance (AC = 83%, PR = 87%, RE = 83%, FS = 83%, AUC-Roc = 92%, MCC = 74%). In testing outcomes, DenseNet and ResNet again achieved the highest scores. DenseNet attained (AC = 92%, PR = 82%, RE = 81%, FS = 82%, AUC-Roc = 94%, MCC = 58%), while ResNet achieved (AC = 91%, PR = 80%, RE = 81%, FS = 81%, AUC-Roc = 94%, MCC = 60%). MLP and CNN both achieved good scores (AC = 89%, PR = 75%, RE = 79%, FS = 77%, AUC-Roc = 91%, MCC = 56%). GRU performed at (AC = 88%, PR = 72%, RE = 64%, FS = 67%, AUC-Roc = 72%, MCC = 21%), CNN-LSTM at (AC = 87%, PR = 68%, RE = 63%, FS = 66%, AUC-Roc =79%, MCC = 45%), and LSTM at (AC = 85%, PR = 58%, RE = 54%, FS = 54%, AUC-Roc = 60%, MCC =5%). Bidirectional LSTM had the lowest performance (AC = 87%, PR = 50%, RE = 50%, FS = 47%, AUC-Roc = 68%, MCC = 19%). Eight DL classification algorithms were applied: MLP, CNN, LSTM, GRU, CNN-LSTM, Bidirectional LSTM, ResNet, and DenseNet. Table 6 displays the results of applying DL with SMOTE Balanced Techniques to the dataset for both training and testing. Figure 10 illustrates a comparative analysis of these DL classifiers based on evaluation metrics. The outcomes were promising, with DenseNet and ResNet achieving the highest accuracy among all classifiers. MLP and CNN showed similar good performance, while GRU and CNN-LSTM demonstrated comparable performance. In contrast, LSTM and Bidirectional LSTM exhibited lower accuracy.

E. RESULTS OF DL WITH ENSEMBLE METHODS TECHNIQUE ONLY

Nine DL classification algorithms were utilized: MLP, CNN, LSTM, GRU, Transformer, Autoencoder, CNN-LSTM, ResNet, and DenseNet.

Nine DL classification algorithms were employed: MLP, CNN, LSTM, GRU, Transformer, Autoencoder, CNN-LSTM, ResNet, and DenseNet. Table 6 shows the results of applying DL with Ensemble Methods Technique to the dataset for both training and testing. DenseNet demonstrated the most effective performance in both training and testing. In terms of training outcomes, DenseNet and MLP achieved the highest scores. DenseNet achieved (AC = 92%, PR = 91%, RE = 92%, FS = 91%, AUC-Roc = 95%, MCC = 59%), while MLP achieved (AC = 91%, PR = 90%, RE = 91%, FS = 91%, AUC-Roc = 96%, MCC = 56%). Autoencoder and ResNet achieved good scores. Autoencoder achieved (AC = 91%, PR = 90%, RE = 91%, FS = 90%, AUC-Roc = 96%, MCC =54%), and ResNet achieved (AC = 91%, PR = 91%, RE = 91%, FS = 89%, AUC-Roc = 95%, MCC = 60%). The remaining algorithms attained the following performance levels: CNN (AC = 90%, PR = 88%, RE = 90%, FS = 87%, AUC-Roc =94%, MCC =42%), LSTM and GRU (AC = 88%, PR = 77%, RE = 88%, FS = 82%, AUC-Roc = 50%, MCC =0%), Transformer (AC = 87%, PR = 84%, RE = 88%, FS = 82%, AUC-Roc = 64%, MCC = 7%), and CNN-LSTM (AC = 88%, PR = 77%, RE = 87%, FS = 82%, AUC-Roc = 50%, MCC = 0%). In testing outcomes, DenseNet and ResNet achieved the highest scores. DenseNet achieved (AC = 91%, PR = 90%, RE = 91%, FS = 91%, AUC-Roc = 94%, MCC =56%), while ResNet

TABLE 5. Results of DL with SMOTE of balanced techniques.

Used Algorithms	Train Data						Test Data					
	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
MLP	0.94	0.94	0.94	0.94	0.99	0.89	0.89	0.75	0.79	0.77	0.91	0.54
CNN	0.94	0.94	0.94	0.94	0.99	0.88	0.89	0.75	0.79	0.77	0.90	0.56
LSTM	0.85	0.86	0.85	0.84	0.89	0.71	0.85	0.58	0.54	0.54	0.60	0.05
GRU	0.90	0.91	0.90	0.90	0.94	0.76	0.88	0.72	0.64	0.67	0.72	0.21
CNN-LSTM	0.89	0.89	0.89	0.89	0.96	0.84	0.87	0.68	0.63	0.66	0.79	0.45
Bidirectional LSTM	0.83	0.87	0.83	0.83	0.92	0.74	0.87	0.50	0.50	0.47	0.68	0.19
ResNet	0.95	0.95	0.95	0.95	0.99	0.90	0.91	0.80	0.81	0.81	0.94	0.60
DenseNet	0.96	0.96	0.96	0.96	0.99	0.90	0.92	0.82	0.81	0.82	0.94	0.58

achieved (AC = 91%, PR = 90%, RE = 91%, FS = 90%, AUC-Roc = 94%, MCC = 56%). The remaining algorithms attained the following performance levels: MLP (AC = 90%, PR = 90%, RE = 90%, FS = 89%, AUC-Roc =93%, MCC =48%), Autoencoder (AC = 90%, PR = 88%, RE = 89%, FS = 89%, AUC-Roc = 93%, MCC = 48%), CNN (AC = 89%, PR = 87%, RE = 89%, FS = 87%, AUC-Roc = 91%, MCC = 38%), Transformer (AC = 88%, PR = 83%, RE = 88%, FS = 82%, AUC-Roc =63%, MCC = 6%), LSTM and GRU (AC = 88%, PR = 77%, RE = 88%, FS = 82%, AUC-Roc = 50%, MCC = 0%), and CNN-LSTM (AC = 87%, PR = 76%, RE = 87%, FS = 81%, AUC-Roc = 50%, MCC = 0%). Figure 11 displays a comparative analysis of the DL classifiers based on evaluation metrics. The outcomes were satisfactory, with DenseNet and ResNet achieving the highest accuracy among all classifications. MLP, Autoencoder, and CNN demonstrated good performance. LSTM and GRU showed similar performance, while Transformer and CNN-LSTM exhibited lower accuracy.

F. RESULTS OF DL WITH ENSEMBLE METHODS AND SMOTE TECHNIQUES

The outcomes of applying DL with Ensemble Methods and SMOTE Technique to the dataset for both training and testing are presented in Table 7.

Table 7 displays the results of applying DL with Ensemble Methods and SMOTE Technique to the dataset for both training and testing. DenseNet demonstrated the most effective performance in both training and testing, achieving the highest scores. In terms of training outcomes, DenseNet achieved(AC = 96%, PR = 96%, RE = 96%, FS = 96%, AUC-Roc = 99%, MCC = 92%). ResNet, MLP, and Autoencoder followed with good scores.ResNet achieved (AC = 95%, PR = 95%, RE = 95%, FS = 95%, AUC-Roc = 99%, MCC = 91%). MLP achieved (AC = 95%, PR = 95%, RE = 95%, FS = 95%, AUC-Roc = 99%, MCC = 89%). Autoencoder achieved (AC = 95%, PR = 95%, RE = 95%, FS = 95%, AUC-Roc = 99%, MCC = 89%). CNN achieved (AC = 94%, PR = 94%, RE = 94%, FS = 94%, AUC-Roc = 92%, MCC =56%), LSTM attained (AC = 86%, PR = 88%, RE = 86%, FS = 85%, AUC-Roc =69%, MCC = 7%), and Transformer exhibited the lowest performance (AC = 84%, PR = 87%, RE = 87%, FS = 84%, AUC-Roc = 66%, MCC = 11%). In testing outcomes, DenseNet again achieved the highest scores (AC = 92%, PR = 92%, RE = 92%, FS = 92%, AUC-Roc =99%, MCC = 64%). ResNet attained (AC = 91%, PR = 91%, RE = 91%, FS = 91%, AUC-Roc = 95%, MCC = 62%), MLP and CNN achieved the same good scores (AC = 90%, PR = 90%, RE = 90%, FS = 90%, AUC-Roc = 93%, MCC = 56%). Autoencoder achieved (AC = 90%, PR = 90%, RE = 90%, FS = 90%, AUC-Roc = 93%, MCC = 89%). LSTM attained (AC = 88%, PR = 83%, RE = 88%, FS = 83%, AUC-Roc =63%, MCC = 7%), and Transformer exhibited the lowest performance (AC = 86%, PR = 81%, RE = 86%, FS = 83%, AUC-Roc = 66%, MCC = 11%). Seven DL classification algorithms were applied: MLP, CNN, LSTM, Transformer, Autoencoder, ResNet, and DenseNet. Figure 12 displays a comparative analysis of the DL classifiers based on evaluation metrics. The outcomes were satisfactory, with DenseNet and ResNet achieving the highest accuracy among all classifications. MLP, Autoencoder, and CNN demonstrated good performance. LSTM had acceptable performance results, while Transformer exhibited lower accuracy.

V. DISCUSSION

The Background section (2) suggests that incorporating artificial intelligence into banking systems is essential for making well-informed decisions about loan eligibility for customers. This paper employs various machine learning (ML) and deep learning (DL) techniques, with promising results presented in subsequent subsections. The training results in Tables 4 and 5 show significant enhancements when models like DenseNet, ResNet, MLP, and CNN are combined with SMOTE and SMOTE TOMEK techniques. In the testing phase, Table 6 demonstrates considerable success with models such as DenseNet, ResNet, MLP, and Autoencoder when using Ensemble Methods techniques. Throughout both

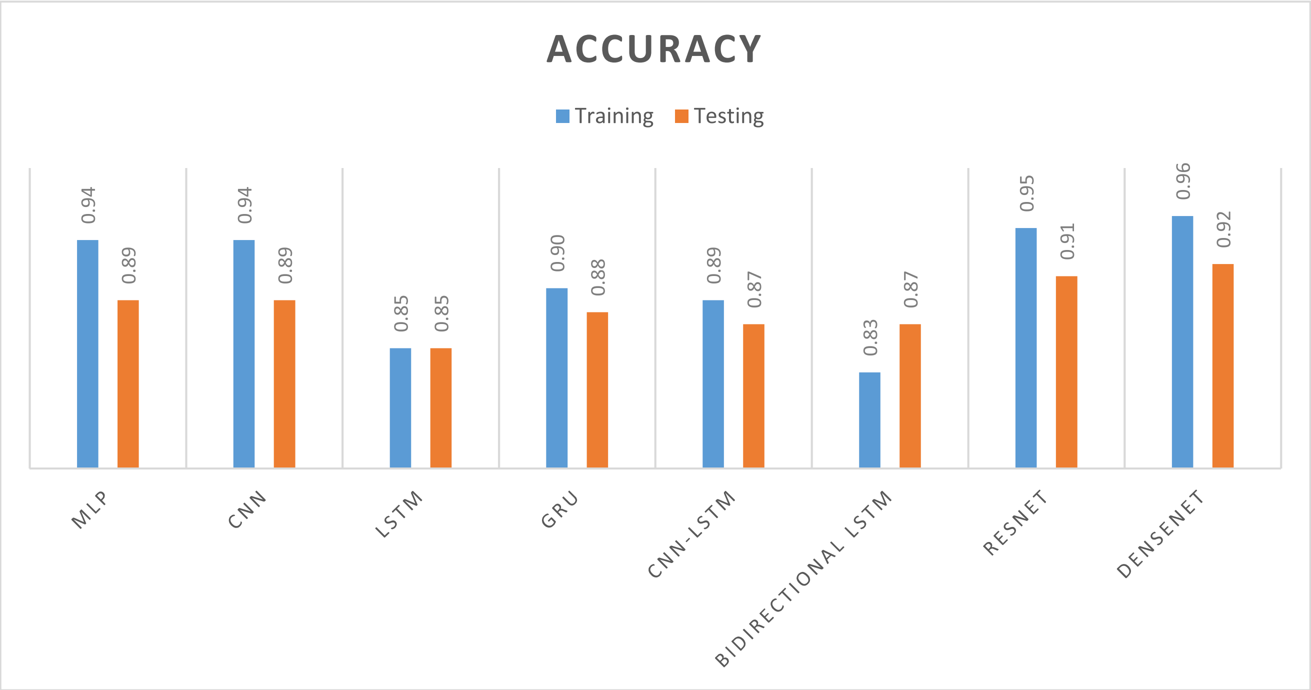

FIGURE 10. Performance comparison chart of DL with SMOTE in balanced techniques.

TABLE 6. Results of DL with ensemble methods technique only.

Used Algorithms	Train Data						Test Data					
	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
MLP	0.91	0.90	0.91	0.91	0.96	0.56	0.90	0.90	0.90	0.89	0.93	0.48
CNN	0.90	0.88	0.90	0.87	0.94	0.42	0.89	0.87	0.89	0.865	0.91	0.38
LSTM	0.88	0.77	0.88	0.82	0.50	0.00	0.88	0.77	0.88	0.82	0.50	0.00
GRU	0.88	0.77	0.88	0.82	0.50	0.00	0.88	0.77	0.88	0.82	0.50	0.00
Transformer	0.87	0.84	0.88	0.82	0.64	0.07	0.88	0.83	0.88	0.82	0.63	0.06
Autoencoder	0.91	0.90	0.91	0.90	0.96	0.54	0.90	0.88	0.89	0.89	0.93	0.48
CNN-LSTM	0.88	0.77	0.87	0.82	0.50	0.00	0.87	0.76	0.87	0.81	0.50	0.00
ResNet	0.91	0.91	0.91	0.89	0.95	0.60	0.91	0.90	0.91	0.90	0.94	0.56
DenseNet	0.92	0.91	0.92	0.91	0.95	0.59	0.91	0.90	0.91	0.91	0.94	0.56

training and testing, ResNet, MLP, and Autoencoder consistently deliver commendable results. The best success is highlighted in Table 7, with models like DenseNet, ResNet, MLP, and Autoencoder when employing ensemble methods and SMOTE techniques. Notably, DenseNet, when used with Ensemble Methods and SMOTE Techniques, consistently outperforms other models in both training and testing, making it the most reliable choice. It is clear that combining DL algorithms with Ensemble Methods, along with SMOTE and SMOTE-TOMEK techniques, leads to more effective results.

The integration of ensemble methods and SMOTE-TOMEK techniques in our Deep Learning models significantly enhances the handling of imbalanced data in loan default prediction. These approaches lead to improved performance metrics, particularly for the minority class, ensuring that the models are both accurate and reliable. By addressing the challenges posed by imbalanced datasets, we can better identify potential loan defaults, ultimately benefiting financial institutions in their risk assessment processes.

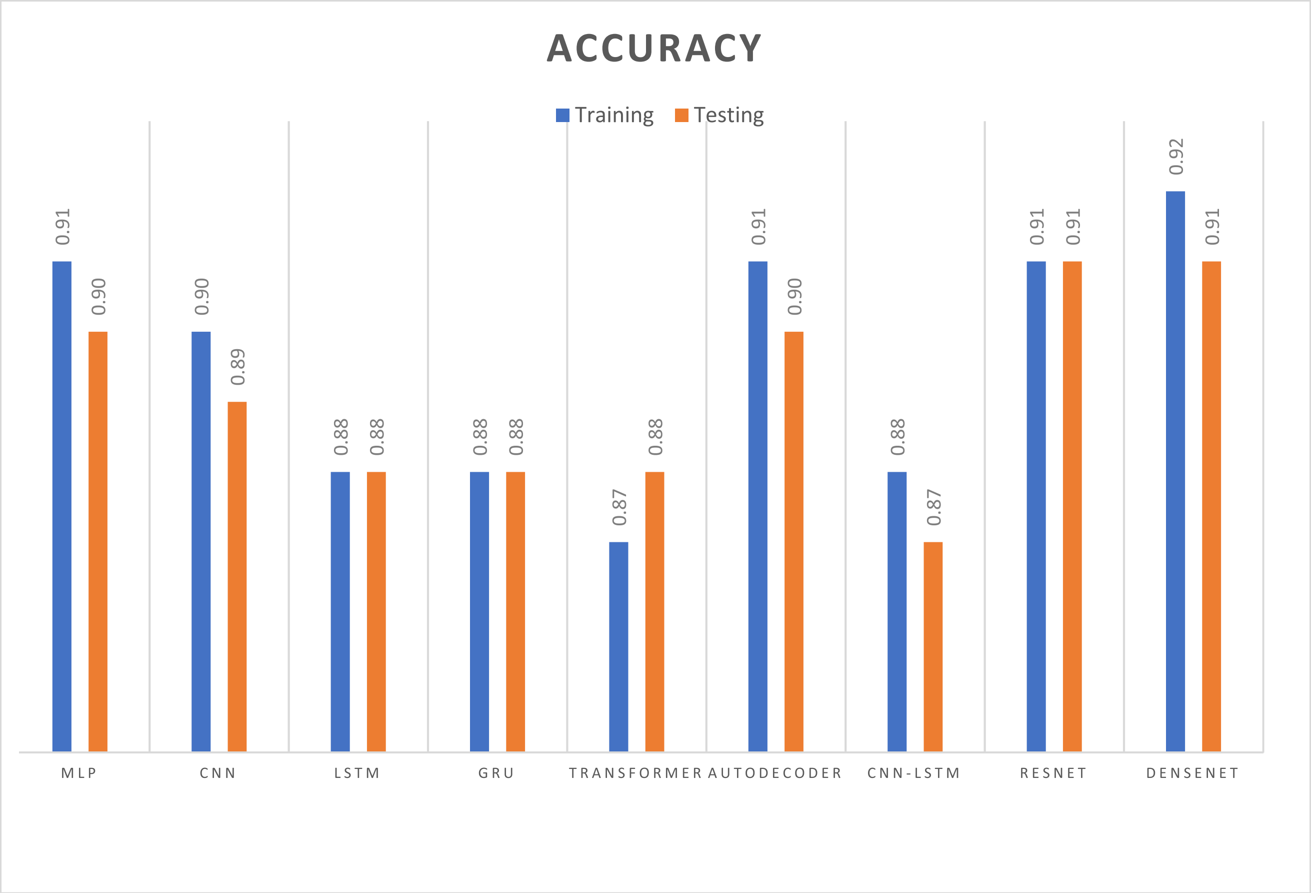

FIGURE 11. Performance comparison chart of DL with ensemble methods technique only.

TABLE 7. Results of DL with ensemble methods and SMOTE techniques.

Used Algorithms	Train Data						Test Data					
	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC	Accuracy	Precision	Recall	F1-Score	AUC-ROC	MCC
MLP	0.95	0.95	0.95	0.95	0.99	0.89	0.90	0.90	0.90	0.90	0.93	0.56
CNN	0.94	0.94	0.94	0.94	0.99	0.89	0.90	0.90	0.89	0.89	0.92	0.56
LSTM	0.86	0.88	0.86	0.85	0.93	0.73	0.88	0.83	0.88	0.83	0.69	0.07
Transformer	0.84	0.87	0.87	0.84	0.91	0.72	0.86	0.81	0.86	0.83	0.66	0.11
Autoencoder	0.95	0.946	0.946	0.945	0.99	0.89	0.90	0.90	0.90	0.90	0.93	0.89
ResNet	0.95	0.95	0.95	0.95	0.99	0.91	0.91	0.91	0.91	0.91	0.95	0.62
DenseNet	0.96	0.96	0.96	0.96	0.99	0.92	0.92	0.92	0.92	0.92	0.95	0.64

This study identifies several specific gaps in traditional methods of loan eligibility assessment that it aims to address:

1. Inefficiency in Processing Applications: Traditional methods often involve manual review processes that are time-consuming and can delay loan approvals, leading to customer dissatisfaction.

2. Inaccuracy in Predicting Defaults: Established eligibility criteria may not appropriately capture the complex behaviors and patterns indicative of potential loan defaults, resulting in a higher risk of approving unsuitable applicants.

3. Limited Adaptability: Traditional methods typically rely on static rules and historical data without the ability to adapt to new patterns in consumer behavior or economic conditions, leading to outdated assessment algorithms.

4. Scalability Issues: As the volume of loan applications grows, traditional methods struggle to scale efficiently, resulting in backlogs and increased operational costs.

Choice of ML and DL Algorithms:

The selection of the specific Machine Learning (ML) and Deep Learning (DL) algorithms in this study was based

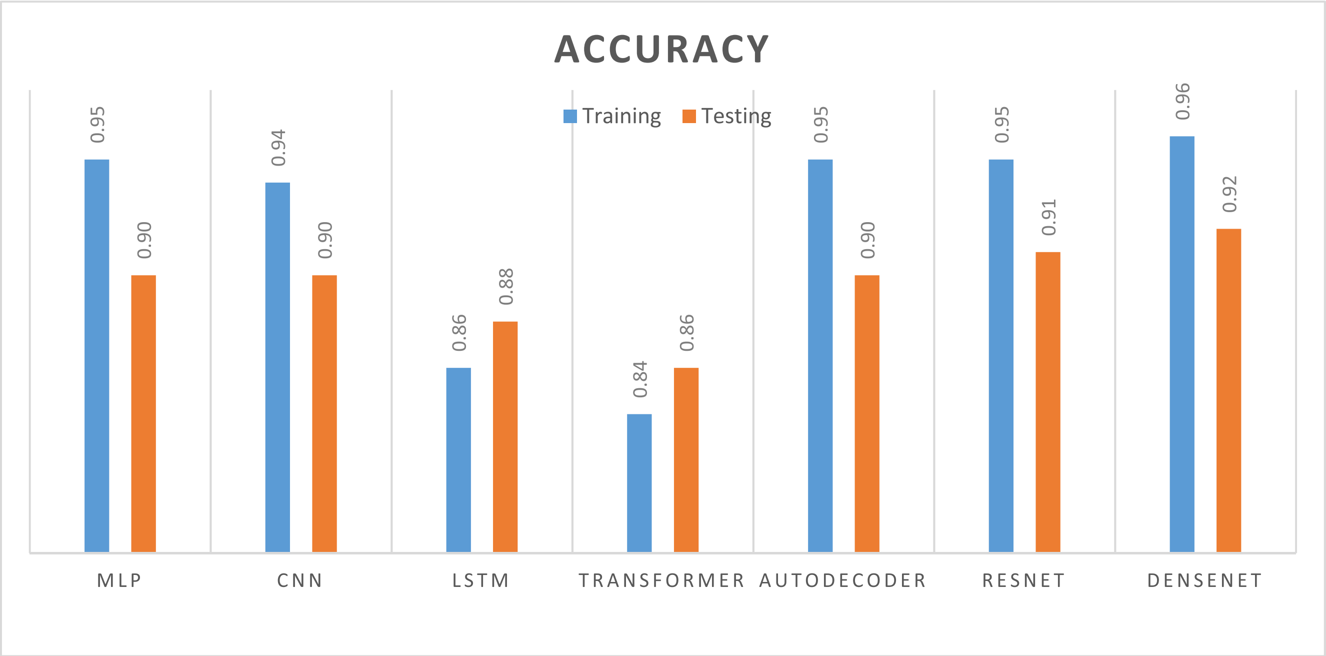

FIGURE 12. Performance comparison chart of DL with ensemble methods and SMOTE techniques.

on their proven effectiveness in handling complex pattern recognition tasks, as well as their ability to improve prediction accuracy across various domains, including financial services.

1. Machine Learning Algorithms:

Gaussian Naive Bayes: Chosen for its simplicity and effectiveness in handling large datasets, particularly when the feature independence assumption reasonably holds.

AdaBoost and Gradient Boosting: Selected for their ability to improve prediction accuracy by combining the outputs of weak classifiers to create a robust predictive model, particularly in dealing with imbalanced datasets.

K Neighbors Classifier: A non-parametric method that offers good performance in a variety of situations without extensive parameter tuning.

Logistic Regression: Included as a benchmark for its interpretability and long-standing application in binary classification problems.

2. Deep Learning Algorithms:

MLP (Multi-Layer Perceptron): Effective for capturing complex nonlinear relationships within the data.

CNN (Convolutional Neural Networks): Although typically used for image data, its capacity for feature extraction makes it suitable for structured data patterns.

LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit): Selected for their ability to handle temporal dependencies within data, which is crucial for understanding borrowers' historical behaviors.

Transformer: Leveraged for its powerful self-attention mechanisms, enabling thorough context analysis in high-dimensional data sets.

Autoencoder: Employed for anomaly detection and data compression, enhancing the feature representation.

ResNet and DenseNet: These architectures are particularly suited for handling deep networks effectively, mitigating issues of vanishing gradients and enabling the modeling of complex relationships within the data.

Comparison to Traditional Methods:

The chosen ML and DL algorithms offer significant advantages over traditional methods, including:

Increased Accuracy: By utilizing advanced techniques for pattern recognition and adaptive learning, the studied algorithms demonstrated superior prediction accuracy compared to traditional criteria-based approaches.

Efficiency: These algorithms can process vast amounts of data in a fraction of the time it takes traditional methods, facilitating real-time decision-making.

Adaptability: Machine Learning and Deep Learning methods can be tuned and retrained with new data, allowing them to remain relevant in changing economic environments and consumer behaviors.

Overall, this study emphasizes the transformative potential of ML and DL techniques in the loan eligibility and risk assessment process, addressing the specific gaps left by traditional methods.

Challenges Encountered and Strategies for Overcoming Them:

1. Data Quality Issues:

Challenge: The quality of data is paramount in predictive modeling. In our study, we faced several data quality issues, including missing values, inconsistencies, and imbalanced datasets. Missing values can lead to biased predictions, while inconsistencies can arise from different data sources or formats. Additionally, the imbalanced nature of the dataset, where the number of non-defaulting applicants significantly outweighed the defaulting ones, posed a challenge for model training.

Strategies:

Data Cleaning: We implemented a rigorous data cleaning process to address missing values through imputation techniques, such as mean/mode imputation for numerical features and the most frequent value for categorical features. This ensured that our dataset was complete and consistent.

Data Normalization: We normalized the data to ensure that all features contributed equally to the model training, which is particularly important for algorithms sensitive to feature scales.

Handling Imbalance: To tackle the class imbalance, we employed the SMOTE (Synthetic Minority Oversampling Technique) and SMOTE-TOMEK techniques. SMOTE helped generate synthetic samples for the minority class (defaulting applicants), while SMOTE-TOMEK further refined the dataset by removing noisy samples, leading to a more balanced training set.

2. Model Performance Challenges:

Challenge: Achieving optimal model performance was another significant challenge. Different algorithms exhibited varying levels of accuracy, precision, recall, and F1-measure. Overfitting was also a concern, particularly with complex models like deep learning architectures, which could perform well on training data but poorly on unseen data.

Strategies:

Cross-Validation: We employed k-fold cross-validation to ensure that our model's performance was robust and not reliant on a single train-test split. This technique allowed us to evaluate the model on multiple subsets of the data, providing a more reliable estimate of its performance.

Hyperparameter Tuning: We conducted extensive hyperparameter tuning using grid search and random search methods to identify the best parameters for each model. This process helped improve model performance and reduce the risk of overfitting.

Ensemble Methods: To enhance predictive accuracy, we utilized ensemble methods, combining the predictions of multiple models. This approach helped mitigate the weaknesses of individual models and improved overall performance.

Regularization Techniques: For models prone to overfitting, such as logistic regression and neural networks, we applied regularization techniques (L1 and L2 regularization) to penalize overly complex models and promote generalization.

Conclusion: By addressing data quality issues and implementing strategies to enhance model performance, we ensured the transparency and reliability of our predictive modeling approach. These efforts not only improved the accuracy of our predictions but also contributed to a more robust understanding of the factors influencing loan defaults.

VI. CONCLUSION

This paper introduces a customer loan prediction method utilizing Artificial Intelligence techniques. By integrating Artificial Intelligence, this approach leverages Machine Learning (ML) and Deep Learning (DL) techniques, employing various classification algorithms to predict potential loan candidates. The study employs five ML classification algorithms—Gaussian Naive Bayes, AdaBoost, Gradient Boosting, K Neighbors Classifier, and Logistic Regression—and eight DL algorithms—MLP, CNN, LSTM, Transformer, GRU, Autoencoder, ResNet, and DenseNet. The incorporation of these algorithms, in conjunction with Ensemble Methods and SMOTE with SMOTE-TOMEK Techniques, significantly improves the results compared to approaches without these methods. To validate these findings, four evaluation metrics—accuracy, precision, recall, and F1-measure—are used. The research highlights DenseNet and ResNet as the most accurate predictive models. By employing predictive modeling, this paper distinguishes high-risk clients from a large pool of loan applicants, resulting in a more effective approach for loan credit approval.

REFERENCES

[1] M. Anand, A. Velu, and P. Whig, "Prediction of loan behaviour with machine learning models for secure banking," *J. Comput. Sci. Eng. (JCSE)*, vol. 3, no. 1, pp. 1–13, Feb. 2022.

[2] L. U. Bhanu and D. S. Narayana, "Customer loan prediction using supervised learning technique," *Int. J. Scientific Res. Publications (IJSRP)*, vol. 11, no. 6, pp. 403–407, Apr. 2021.

[3] D. K. Malhotra, K. Malhotra, and R. Malhotra, "Evaluating consumer loans using machine learning techniques," in *Applications of Management Science*. Bingley, U.K.: Emerald Publishing Limited, 2020, pp. 59–69.

[4] J. Chen, A. L. Katchova, and C. Zhou, "Agricultural loan delinquency prediction using machine learning methods," *Int. Food Agribusiness Manage. Rev.*, vol. 24, no. 5, pp. 797–812, Jul. 2021.

[5] S. Zahi and B. Achchab, "Modeling car loan prepayment using supervised machine learning," *Pro. Comput. Sci.*, vol. 170, pp. 1128–1133, Jan. 2020.

[6] S. Sreesouthry, A. Ayubkhan, M. M. Rizwan, D. Lokesh, and K. P. Raj, "Loan prediction using logistic regression in machine learning," *Ann. Romanian Soc. Cell Biol.*, pp. 2790–2794, 2021.

[7] T. Ndayisenga, "Bank loan approval prediction using machine learning techniques," Master's dissertation, 2021.

[8] S. I. Serengil, S. Imece, U. G. Tosun, E. B. Buyukbas, and B. Koroglu, "A comparative study of machine learning approaches for non performing loan prediction with explainability," *Int. J. Mach. Learn. Comput.*, vol. 12, no. 5, pp. 1102–1110, 2022.

[9] A. Shinde, Y. Patil, I. Kotian, A. Shinde, and R. Gulwani, "Loan prediction system using machine learning," in *Proc. ITM Web Conf.* Les Ulis, France: EDP Sciences, 2022, p. 3019.

[10] J. Smith and R. Doe, "Predictive modeling of loan default: A comparative analysis of traditional machine learning techniques," *J. Financial Technol.*, vol. 15, no. 3, pp. 123–135, 2022.

[11] M. Johnson and A. Chang, "A deep learning approach to modeling loan repayment risk," *Int. J. Data Sci.*, vol. 8, no. 2, pp. 45–60, 2023.

[12] K. Lee, T. Brown, and L. Zhang, "Ensemble learning in financial risk assessment: A case study," *J. Finance Risk Perspect.*, vol. 20, no. 1, pp. 78–90, 2021.

[13] S. Patel and R. Kumar, "Comparative study on machine learning and deep learning models for credit risk assessment," *Appl. Soft Comput.*, 2022.

[14] P. K. Donepudi, "Machine learning and artificial intelligence in banking," *Eng. Int.*, vol. 5, no. 2, pp. 83–86, 2017.

[15] M. A. Sheikh, A. K. Goel, and T. Kumar, "An approach for prediction of loan approval using machine learning algorithm," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 490–494.

[16] M. B. Nag and F. Ahmad Malik, "Data analysis and interpretation," in *Repatriation Management and Competency Transfer in a Culturally Dynamic World*. Cham, Switzerland: Springer, 2023, pp. 93–140.

[17] D. Dansana, S. G. K. Patro, B. K. Mishra, V. Prasad, A. Razak, and A. W. Wodajo, "Analyzing the impact of loan features on bank loan prediction using random forest algorithm," *Eng. Rep.*, vol. 6, no. 2, 2023, Art. no. e12707.

[18] V. Singh, A. Yadav, R. Awasthi, and G. N. Partheeban, "Prediction of modernized loan approval system based on machine learning approach," in *Proc. Int. Conf. Intell. Technol. (CONIT)*, Jun. 2021, pp. 1–4.

[19] P. Kumar and N. Sood, "Controller-driven event-based network heat map visualisation," Tech. Rep., 2023.

[20] S. D. R. Subburaj, V. V. Rengarajan, and S. Palaniswamy, "Transfer learning based image classification of diseased tomato leaves with optimal fine-tuning combined with heat map visualization," *J. Agricult. Sci.*, vol. 29, no. 4, pp. 1003–1017, 2023.

[21] C. Hao and S. Adsavakulchai, "The use of machine learning algorithms for bank loan prediction," *Eur. Econ. Lett. (EEL)*, vol. 13, no. 3, pp. 735–741, 2023.

[22] J. C. Alejandrino, J. J. P. Bolacoy, and J. V. B. Murcia, "Supervised and unsupervised data mining approaches in loan default prediction," *Int. J. Elect. Comput. Eng. (IJECE)*, vol. 13, no. 2, p. 1837, Apr. 2023.

[23] Y. Dasari, K. Rishitha, and O. Gandhi, "Prediction of bank loan status using machine learning algorithms," *Int. J. Comput. Digit. Syst.*, vol. 14, no. 1, pp. 139–146, Jul. 2023.

[24] S. Reddy, "Loan approval prediction using decision tree," Tech. Rep.

[25] D. Breskuvienė and G. Dzemyda, "Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions," *Int. J. Comput. Commun. Control*, vol. 18, no. 3, pp. 1–17, May 2023.

[26] A. Seveso, A. Campagner, D. Ciucci, and F. Cabitza, "Ordinal labels in machine learning: A user-centered approach to improve data validity in medical settings," *BMC Med. Informat. Decis. Making*, vol. 20, no. S5, pp. 1–14, Aug. 2020.

[27] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Machine learning and deep learning for phishing email classification using one-hot encoding," *J. Comput. Sci.*, vol. 17, no. 7, pp. 610–623, Jul. 2021.

[28] D. U. Ozsahin, M. Taiwo Mustapha, A. S. Mubarak, Z. Said Ameen, and B. Uzun, "Impact of feature scaling on machine learning models for the diagnosis of diabetes," in *Proc. Int. Conf. Artif. Intell. Everything (AIE)*, Aug. 2022, pp. 87–94.

[29] A. S. Desuky and S. Hussain, "An improved hybrid approach for handling class imbalance problem," *Arabian J. Sci. Eng.*, vol. 46, no. 4, pp. 3853–3864, Apr. 2021.

[30] C. Kumari, M. Abulaish, and N. Subbarao, "Using SMOTE to deal with class-imbalance problem in bioactivity data to predict mTOR inhibitors," *Social Netw. Comput. Sci.*, vol. 1, no. 3, pp. 1–7, May 2020.

[31] B. Silva, R. Silveira, M. Silva Neto, P. Cortez, and D. Gomes, "A comparative analysis of undersampling techniques for network intrusion detection systems design," *J. Commun. Inf. Syst.*, vol. 36, no. 1, pp. 31–43, 2021.

[32] R. Tang, L. De Donato, N. Bešinović, F. Flammini, R. M. P. Goverde, Z. Lin, R. Liu, T. Tang, V. Vittorini, and Z. Wang, "A literature review of artificial intelligence applications in railway systems," *Transp. Res. C, Emerg. Technol.*, vol. 140, Jul. 2022, Art. no. 103679.

[33] S. Y. Sen and N. Özkurt, "Convolutional neural network hyperparameter tuning with Adam optimizer for ECG classification," in *Proc. Innov. Intell. Syst. Appl. Conf. (ASYU)*, Oct. 2020, pp. 1–6.

[34] G. Varoquaux and O. Colliot, "Evaluating machine learning models and their diagnostic value," in *Machine Learning for Brain Disorders*, 2023, pp. 601–630.

[35] S. S. Bafjaish, "Comparative analysis of Naive Bayesian techniques in health-related for classification task," *J. Soft Comput. Data Mining*, vol. 1, no. 2, pp. 1–10, 2020.

[36] B. T. Pham, M. D. Nguyen, T. Nguyen-Thoi, L. S. Ho, M. Koopialipoor, N. Kim Quoc, D. J. Armaghani, and H. V. Le, "A novel approach for classification of soils based on laboratory tests using AdaBoost, tree and ANN modeling," *Transp. Geotechnics*, vol. 27, Mar. 2021, Art. no. 100508.

[37] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021.

[38] N. Sawant and D. R. Khadapkar, "Comparison of the performance of GaussianNB algorithm, the k neighbors classifier algorithm, the logistic regression algorithm, the linear discriminant analysis algorithm, and the decision tree classifier algorithm on same dataset," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 1654–1665, Dec. 2022.

[39] A. Das, "Logistic regression," in *Encyclopedia of Quality of Life and Well-Being Research*. Cham, Switzerland: Springer, 2021, pp. 1–2.

[40] X. Wang, Y. Zhao, and F. Pourpanah, "Recent advances in deep learning," *Int. J. Mach. Learn. Cybern.*, vol. 11, pp. 747–750, Jan. 2020.

[41] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[42] X. Zhang, X. Wang, H. Chen, D. Wang, and Z. Fu, "Improved GWO for large-scale function optimization and MLP optimization in cancer identification," *Neural Comput. Appl.*, vol. 32, no. 5, pp. 1305–1325, Mar. 2020.

[43] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, Mar. 2021.

[44] Y. Wang, M. Wang, Y. Pan, and J. Chen, "Joint loan risk prediction based on deep learning-optimized stacking model," *Eng. Rep.*, vol. 6, no. 4, 2023, Art. no. e12748.

[45] A. Moghar and M. Hamiche, "Stock market prediction using LSTM recurrent neural network," *Pro. Comput. Sci.*, vol. 170, pp. 1168–1173, Jan. 2020.

[46] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. NIPS*, 2021, pp. 15908–15919.

[47] B. Behinaein, A. Bhatti, D. Rodenburg, P. Hungler, and A. Etemad, "A transformer architecture for stress detection from ECG," in *Proc. Int. Symp. Wearable Comput.*, Sep. 2021, pp. 132–134.

[48] K. Zhang, N. Chen, J. Liu, and M. Beer, "A GRU-based ensemble learning method for time-variant uncertain structural response analysis," *Comput. Methods Appl. Mech. Eng.*, vol. 391, Mar. 2022, Art. no. 114516.

[49] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," *J. Netw. Comput. Appl.*, vol. 177, Mar. 2021, Art. no. 102942.

[50] P. Li, Y. Pei, and J. Li, "A comprehensive survey on design and application of autoencoder in deep learning," *Appl. Soft Comput.*, vol. 138, May 2023, Art. no. 110176.

[51] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, Sep. 2018.

[52] F. He, T. Liu, and D. Tao, "Why ResNet works? Residuals generalize," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5349–5362, Dec. 2020.

[53] Y. Zhu and S. Newsam, "DenseNet for dense flow," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 790–794.

[54] N. Hasan, Y. Bao, A. Shawon, and Y. Huang, "DenseNet convolutional neural networks application for predicting COVID-19 using CT image," *Social Netw. Comput. Sci.*, vol. 2, no. 5, p. 389, Sep. 2021.

[55] Ž. Đ. Vujović, "Classification model evaluation metrics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 599–606, 2021.

[56] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, Dec. 2020.

[57] G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in *Proc. Comput. Sci. On-Line Conf.* SpringerCham, Switzerland: Springer 2023, pp. 15–25.

[58] C. Cao, D. Chicco, and M. M. Hoffman, "The MCC-F1 curve: A performance evaluation technique for binary classification," 2020, *arXiv:2006.11278*.