Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 6, 2025

NEXT-GEN CUSTOMER RETENTION: A STACKED ENSEMBLE MODEL FOR CHURN PREDICTION

Muhammad Shahan Ibad[1], Syed Noor Hussain Shah[2], Omar Bin Samin[3], Sumaira Johar*[4]

[1,2,3,*4]*School of Computer Science & IT, Institute of Management Sciences (IMSciences), Peshawar, Pakistan*

[1]shaniims2022@gmail.com, [2]asfibang04@gmail.com, [3]omar.samin@imsciences.edu.pk,
*[4]sumaira.johar@imsciences.edu.pk

[1]https://orcid.org/0009-0003-2719-313X, [2]https://orcid.org/0009-0006-9658-2747,
[3]https://orcid.org/0000-0001-8017-9845, *[4]https://orcid.org/0000-0002-9033-1596

DOI: https://doi.org/

Keywords
Customer churn prediction, Ensemble stacking model, Machine learning classifiers, Proactive customer retention, Latency Aware Accuracy Index (LAAI)

Article History
Received on 11 May 2025
Accepted on 11 June 2025
Published on 18 June 2025

Copyright @Author
Corresponding Author: *
Sumaira Johar

Abstract

Customer churn is one major problem in the telecom industry, requiring efficient and effective predictive models for proactive customer retention. Much work has already been achieved in this direction, but most studies so far have focused mainly on individual classifiers. While these models perform well in many areas, they have their own set of weaknesses. These include computational inefficacy, susceptibility to dataset imbalance, and inability to learn from subtle relations. Therefore, they tend to be insufficient in optimizing the trade-off between computational cost and accuracy. Existing methods tend to be insufficient with high-dimensional datasets, vulnerable to overfitting, or non-generalizable across telecom datasets. This work proposes an Ensemble Stacking model that is capable of overcoming these weaknesses. The proposed model consists of a set of base learners, Random Forest, Naïve Bayes, and K-Nearest Neighbors that are responsible for learning various patterns in the dataset. The base models make predictions, which in turn feed a single meta-model, Logistic Regression, that learns from their predictions to make the final prediction. The results reveal that the proposed model is capable of generating excellent accuracy with acceptable latency, outperforming all individual classifiers. Its superior latency-aware accuracy Index (LAAI) score also validates the fact that it is highly robust and adaptable, making it a very effective solution for real-world prediction problems.

INTRODUCTION

In the modern world, practically everyone now considers phone and internet services to be essential. Every day, individuals communicate with others, watch videos, send messages, and work online using their phones and the internet. These businesses, which include internet and mobile providers, are referred to as telecom companies. To make money and remain in business, telecom firms need consumers. Satisfied clients are crucial to the growth and success of telecom companies. However, customers may discontinue using a business's services. Perhaps the costs are too exorbitant, the internet is too slow, or they came upon a better offer elsewhere. As a result, the business loses that client. We call this customer churn. For telecom firms, customer attrition is a major issue. Because the business loses money when a customer quits, churn has a direct impact on revenue. Retaining existing consumers is far less expensive for the business than acquiring new ones. According to a survey [1] 30–35% of customers unsubscribed from their telecom industry per year after COVID.

Understanding why customers cease using a service is the first step in managing customer churn. High costs, poor network quality, and appealing offers from competitors can all lead to turnover. Businesses must identify clients who are likely to quit and implement retention initiatives to prevent attrition. Telecom corporations study a variety of user data, including billing history, complaints, internet usage, and call frequency, to uncover hidden patterns that explain turnover. Machine learning allows us to detect hidden tendencies [2].

The aim of the research paper is to build an ML model that predicts telecom customer attrition through actual customer data. By identifying important churn-causing elements, this study hopes to provide actionable client retention techniques that can help telecom organizations save money, improve customer pleasure, and prosper in a competitive market. Previous research has achieved substantial advances in telecom churn prediction, emphasizing the impact of contract type, monthly prices, and service type. However, early systems frequently lacked advanced feature engineering, ensemble methods, or scalability, which limited their practical application. This work fills these gaps by integrating Naive Bayes, Random Forest, Logistic Regression, and KNN in an Ensemble Stacking to obtain both high accuracy and computational frugality while ensuring interpretability. It focuses on scalable methods and feature interactions to make the model fit in real-world contexts by various telecom organizations.

Customer churn has an effect on earnings in telecommunication companies and stability in the market. Current practices are computationally efficient yet not interpretable and not scalable to apply in practice. Actionable outcomes are limited by not taking into consideration the interplay between features and trends over time. This paper suggests filling these gaps by taking an ensemble stacking approach to balance scalability, interpretation, and effectiveness in practice by combining Random Forest, KNN, Logistic Regression, and Naive Bayes.

LITERATURE REVIEW

Machine learning algorithms have been employed widely in order to identify the key drivers for customer churn and to develop prediction models for telecom operators. This study utilizes both an openly available dataset, churn-bigml, and a company-specific dataset from a South Asian GSM telecom operator. The approach is successful in detecting churned customers as well as in detecting the underlying reasons for their churn. Churn prediction is a critical activity in the highly competitive telecom market since it is much more economical to retain existing customers than acquire new ones. The study employs Random Forest classification with 88.63% accuracy in predicting churn and K-means clustering in order to segment the customers based on their likelihood of churning. These results enable telecom operators to design focused retention policies that enhance customer retention and maximize profitability. The study establishes that machine learning has the capability in reducing churn rates significantly and in improving the overall customer experience [3]. Extending these findings, another study work in churn prediction utilizes the WA_Fn-UseC_ dataset (7,000+ records) from Kaggle, with demographic, usage, and account information. The workflow includes preprocessing, exploratory data analysis (EDA), and model building with various machine learning algorithms such as Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Linear Regression, Lasso, Ridge, and AdaBoost Regressor. In order to address class imbalance, SMOTEENN upsampling is applied After tuning the hyperparameters, Random Forest Classifier is the best performing model with Precision 0.9649, Recall 0.9635, and F1-score 0.9642. The study identifies the key drivers for churn, enabling telecom enterprises in developing evidence-driven retention policies [4]. Taking this research, a step further, another research is based on customer churn prediction using machine learning algorithms on a Maven Analytics dataset containing 7,043 records and 38 attributes, including customer activity data and a churn label [5]. The work is performed according to the CRISP-DM process, including problem definition, preprocessing of the data, model implementation, evaluation, and interpretation. It uses the following machine learning techniques: Logistic Regression, K-Nearest Neighbors (KNN), Naïve Bayes, Decision Tree, and Random Forest, with the latter achieving the highest accuracy at 86.94%, AUC at 0.95, sensitivity at 0.8547, and specificity at 0.8839. For making the results more interpretable and explainable, Explainable AI (XAI) techniques such as Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanation (SHAP) are applied. SHAP analysis identifies 'Contract', 'Number of Referrals', 'Tenure in Months', 'Monthly Charge', and 'Online Security' as the most contributing factors in customer churn. The overall findings from such studies prove the effectiveness of ensemble learning methods, in this case, Random Forest, in predicting churn. Application of XAI methods also raises the level of explain ability in the model, enabling telecom operators to derive meaningful information from customer activity. These findings enable firms to formulate personalized retention policies, reducing churn and customer relations improvement [5].

Similarly, another study on Customer churn prediction is a critical business approach that maximizes customer retention and minimizes revenue loss. Literature has emphasized that all features do not contribute proportionally in prediction, and feature selection is, therefore, important to maximize model performance. An experiment attempted sequential feature selection techniques with the Telco Customer Churn dataset, demonstrating that early detection of likely churners allows proactive measures, ultimately lowering churn rates and keeping customers. The experiment suggested a model with features suitable in an attempt to optimize against churn. Results were such that the base Naïve Bayes with feature selection was at a level of 65.98%. But with the use of four feature selection techniques Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), Sequential Floating Forward Selection (SFFS), and Sequential Floating Backward Selection (SBFS) the level was greatly enhanced, with SBS and SBFS at a level of 71.33% with 19 features. These findings establish the importance of feature selection in developing efficient churn prediction models to help telecom operators adopt specific retention policies [6]. Subsequent studies thereafter suggested a more sophisticated methodological framework that included baseline model estimation, feature selection with parameter tuning, and class imbalance control with the use of SMOTE Tomek and SMOTE ENN. Using the Orange S.A. Telecom dataset, this study applied Pearson's Correlation, SFS, and SBS in feature selection, further reducing the prediction framework. Findings demonstrated that the Radial Basis Function (RBF) SVM with parameter optimization and training using a dataset that was balanced using SMOTE ENN was the most effective, with a rate of 99% and an F1-score of 98.88%. This was far superior to current models and highlighted the importance of feature selection and balance in the dataset in churn prediction. The study concluded that the combination of advanced resampling techniques with kernel SVM significantly enhances predictive capability, with practical implications for customer retention optimization in telecommunication. Further studies can be in the area of more hyperparameter tuning and other machine learning algorithms for additional predictive capability enhancement [7].

Building on these insights, another research study focuses on the Customer churn prediction model utilizing the K-Nearest Neighbors (KNN) algorithm and the Pearson Correlation Function for enhanced prediction precision for telecom operators. The Telecom Customer Churn dataset was utilized, with training and testing sets distributed in a proportion of 70/30 to evaluate model efficacy. The primary objective was the application of KNN and Pearson Correlation in telecommunication, as KNN is effective in classifying information without prior knowledge about the distribution. The model is designed for telecom operators for the prediction of churn-at-risk subscribers so that proactive retention is possible. With training accuracy at 80.45% and testing accuracy at 97.78%, the researchers demonstrated that the KNN algorithm is effective in predicting customer churn. Telecom operators can prevent revenue loss and improve customer retention by such prediction models [8].

In addition, A machine learning approach for customer churn prediction in the telecommunication industry, with customer retention in mind in the face of increasing market competition [9]. The study aims to suggest a precise and efficient churn prediction model using advanced machine learning techniques. A six-step process is suggested for model construction. Data preprocessing is emphasized in the first two steps, and feature selection is implemented in the third step using the Gravitational Search Algorithm (GSA). The dataset, although unnamed, consists of approximately 7,000 records with 21 features and is distributed in training 80% and testing 20%. In the fourth step, the various classifiers, namely Decision Tree, Support Vector Machine (SVM), Logistic Regression, Naïve Bayes, and Random Forest are implemented. The study evaluates the model performance using a confusion matrix and AUC plots, with AdaBoost and XGBoost classifiers having the maximum accuracy at 81.71% and 80.8%, respectively. The two models also achieve the maximum AUC score of 84%, outperforming the rest of the algorithms. Other findings include Logistic Regression: 80.45%, Decision Tree: 80.14%, KNN Classifier: 79.64%, Random Forest: 78.04%, AdaBoost (Extra Tree): 81.14%, Random Forest (AdaBoost): 81.21%, SVM (AdaBoost): 74.07%, SVM (Poly Kernel): 80.21%, SVM (Linear Kernel): 79.14%, Naïve Bayes (Gaussian): 77.07%, CatBoost: 81.8%. The findings support that machine learning algorithms provide a successful and profitable solution for telecommunication operators in customer churn management. With accurate identification of probable churners, firms can implement specific retention policies, thus gaining customer loyalty and profitability.

Furthermore, The Churn Prediction in the Telecommunications Industry Using SVM [10]. The study provides an introduction to predicting customer churn in the mobile cellular market by an SVM algorithm with four different kernel functions. This dataset used in this work is known as Churn Data Set to design prediction models, utilizing four different kernel functions in the SVM algorithm, particularly with polynomial and radial basis function kernels. This work to predict customer churn in the mobile cellular market confirmed that the polynomial kernel function achieved the highest overall accuracy at 88.56% RBF while linear kernel functions also produced an exceptional performance to achieve around 80% in predicting churners. This work concludes that these models in particular those that employ RBF, linear, and polynomial kernel functions can be an invaluable tool to mobile cellular companies. This provides an efficient method to identify customers who have an increased potential to churn, thus allowing targeted retentions to offset customers who depart in an intense market.

Moreover, The Naïve Bayes and Discretization methods in subscriber churn prediction in the telecom industry. The primary objective of the study was to develop a model that would be capable of detecting future leavers, highlighting the substantial cost difference between customer retention and customer acquisition. The study provided a comparative evaluation of Equal-Width Discretization (EWD) and K-Means clustering, both with a Naïve Bayes classifier, to enhance churn prediction. The study concluded that while K-Means was excellent in the detection of future leavers, EWD models were generally superior. Specifically, using EWD with the "total day minutes" attribute with seven clusters achieved a 65.69% rate. In addition, the combination of Naïve Bayes with K-Means was found to be a promising path, especially with further tuning in discretization and class imbalance. Following this, a study [11] investigates customer churn prediction with a more extensive set of data mining approaches. With a publicly available dataset from French telecom company Orange, the study analyzes more than 3,000 customer records with 20 features, such as total night and day minutes, international plan subscriptions, customer service calls, and voicemail plans. The dataset feature "Churn" is subscription cancellation. The study utilizes classification algorithms in the WEKA data mining software to predict churn, testing Naïve Bayes, Random Forest, Neural Networks, and Decision Trees. Results reveal that the Decision Tree algorithm has the highest rate of prediction 94.03%, with better results than with other methods. Naive Bayes is most efficient at 0.03 seconds but less precise with a correctness rate of 88.24%. Random Forest is correct in 91.06% with a time consumption of 1.06 seconds, and Neural Network is correct in 90.16% but is the slowest 53.9 seconds. The conclusion is that Decision Trees provide the most efficient solution through their balance between cost and accuracy. In light of the findings in the study [10], which studied the impact of discretization techniques in combination with Naïve Bayes, the study [11] extends the study to a wider variety of classification algorithms, concluding that Decision Trees provide a more efficient, more accurate solution for telecommunications customer churn prediction. While earlier studies established the promise that Naïve Bayes with discretization was, this study demonstrates that Decision Trees outperform alternative methods in real application.

Additionally, Customer churn prediction in landline services within the telecom industry by introducing new feature sets and comparing multiple machine learning algorithms [12]. This work utilized a novel dataset with 827,124 customers, collected by an Irish telecom company, with equal splits for training and testing sets. There were 13,562 churners and 400,000 non-churners in each subset, with the customers being described by 738 attributes, e.g., demographics, account status, orders, calls, complaints, and billing. Steps in the methodology included preprocessing in the form of attribute extraction and normalization to ensure uniformity in the data. To predict churn, seven learning algorithms were utilized, e.g., Logistic Regression, Decision Tree, Naïve Bayes, Support Vector Machine (SVM), Multilayer Perceptron (MLP), Linear Classifiers, and Evolutionary Data Mining Algorithm (DMEL). Accuracy, true positives (TP) and false positives (FP) rates, Area Under the Curve (AUC), and Receiver Operating Characteristic (ROC) plots were utilized for evaluation, serving as the platform for further advancements in churn prediction models [12]. Following this, later studies focused on enhancing prediction accuracy by utilizing deep learning techniques [13]. In contrast with the previous study, which was predominantly machine learning model-based, the introduction of ChurnNet, a deep learning-based model, significantly improved predictive performance. In this, the utilization was made of 1D Convolutional Neural Networks (CNNs), residual learning, squeeze and excitation blocks, and spatial attention mechanisms in order to improve the model in detecting churn. Three publicly available datasets, including IBM Telco Dataset, Churn-in-Telecom Dataset, and Churn-data-UCI Dataset, were used to train and cross-validate ChurnNet to tackle class imbalance by utilizing advanced techniques such as SMOTE, SMOTETomek, and SMOTEEN. Optimization was further augmented by the utilization of the application of 10-fold cross-validation and hyper parameter tuning. Compared with the conventional machine learning algorithms such as Logistic Regression, SVM, LSTM, and GRU, ChurnNet produced superior accuracy, recall, precision, F-measure, AUC, and MCC values. The results established that ChurnNet attained 95.59% in IBM Telco Dataset, 96.94% in Churn-in-Telecom Dataset, and 97.52% in Churn-data-UCI Dataset, far better than existing models [13]. Furthering this study, a study suggested a Ratio-based data balance method to reverse the impact of imbalanced churn datasets [14]. The techniques employed were data extraction, preprocessing, handling class imbalance, application of machine learning model, and accuracy, precision, recall, and F-score evaluation. Multiple machine learning models were attempted, ranging from individual models such as Perceptron, Multi-Layer Perceptron (MLP), Naïve Bayes, Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree to ensemble models such as Gradient Boosting and Extreme Gradient Boosting (XGBoost). The work attempted the effectiveness of varying balance ratios 90:10, 80:20, 75:25, 65:35, and 50:50 comparing the proposed method with the conventional Over-Sampling and Under-Sampling methods. Results demonstrated that ensemble algorithms, particularly XGBoost, outperformed individual algorithms consistently. The suggested balancing method with optimal results was from the 75:25 ratios, with an achieved rate of 89.60% with XGBoost, with superior precision of 76.04%, recall of 62.82%, and F1-score 61.63%. These findings highlight the negative impact of imbalanced datasets on predictive correctness and demonstrate the crucial role played by balancing the data in enhancing churn prediction. Further, findings suggest that more samples in the data would help enhance model efficacy, contributing towards ongoing innovation in churn prediction methods [14].

Additionally, Telecommunications customer churn prediction (CCP) has been studied extensively, particularly in the context of how uncertain samples contribute to prediction performance [15]. A lot of existing work has focused on whether sample proximity to the majority class can be leveraged to enhance churn prediction. One such study specifically concentrated on voluntary churn and examined whether incorporating uncertain samples could improve predictive performance. Using the Naïve Bayes classifier, this research classified non-churn and churn customers while evaluating performance through recall, F1-score, accuracy, and precision. Three datasets were employed, containing 3,333, 7,043, and 5,783 samples, with churn rates of 14.49%, 73.46%, and 87.84%, respectively. The preprocessing phase involved the discretization of the numerical attributes in ten sets (0–9) and the conversion of categorical attributes into numerical values. The Manhattan distance was utilized by the study in ranking samples and in computing an ascending order of sums of distances, before splitting the dataset into training and testing sets. The testing sets were further split into lower-distance testing (LDT) samples and upper-distance testing (UDT) samples. The training set was utilized in training the Naïve Bayes classifier and separately testing it with the LDT and UDT samples, with the test set size increased by 100 samples in each iteration. The findings were that prediction with the use of LDT samples significantly outperformed prediction with the use of UDT samples, with the three datasets experiencing improvement in performance by LDT samples by 5.91%, 5.60%, and 4.20%. Prediction with the use of UDT samples was, by contrast, generally stable with marginal increments of 30%, 80%, and 81%. These results emphasize the crucial role of uncertain samples, particularly LDT samples, in optimizing CCP models for the telecom industry. In addition, a Bayesian binomial test confirmed the null hypothesis, meaning that the decisions made by the classifier were equivalent to random guesses in specific cases. Follow-up studies further advanced churn prediction by developing a systematic framework for selecting the most suitable statistical method [16]. With a combination of expert judgments and multiple performance metrics, this study applied a mixed-methods approach consisting of a systematic review, experimental testing, and a multi-criteria decision-making (MCDM) technique in the Best-Worst Method (BWM) format. The literature review identified six widely used statistical methods Decision Tree, Logistic Regression, Support Vector Machine (SVM), Random Forest, Gradient Boosted Trees (GBT), and Multi-layer Perceptron Neural Network alongside twelve relevant performance metrics. These methods were compared empirically with a dataset that included 20,000 customer records, training, and testing sets of 80% and 20%, respectively. Model parameters were tuned for accuracy and AUC, and learning curves were plotted to investigate performance at varying sizes for the training set. Expert opinion was also incorporated through the use of the BWM method, with pairwise comparison by 35 academics and professionals in the field of data science to determine the relative importance of the following five selected measures: Accuracy, Precision, AUC, Ease of Interpretation, and Fastness. Aggregated performance for each statistical method was determined by the application of an additive value function, producing an overall ranking. Results indicated that the Decision Tree model was the overall winner, performing extremely well in terms of accuracy and ease of interpretation. Final rankings were: Decision Tree 81.11, Random Forest 75.34, Logistic Regression 74.03, Gradient Boosted Trees 71.57, Support Vector Machine 67.70, Neural Network 65.00 Although ensemble methods such as Random Forest and Gradient Boosted Trees provided better AUC values, the Decision Tree model provided the most balanced performance across the measures. This work re-emphasizes the importance of the use of more than the standard measures in the evaluation of churn prediction models, with the trend towards the use of the multi-criteria method in making the optimal model choice for real-world applications. Following on from the work that considered the use of uncertain samples in churn prediction, this work provides an extended framework in the selection of stable predictive models.

In Addition, Cross-company customer churn prediction (CCCP) in the telecommunication context, with particular focus placed on how varying methods of data transformation (DT) can impact model performance [17]. The authors evaluate the effectiveness of Box-Cox, Z-score, rank, and log transforms in enhancing CCCP prediction. The experiment is done with a source dataset consisting of 18,000 samples and a target dataset consisting of 3,333 samples, trained with five machine learning classifiers, that is, Gradient Boosted Tree (GBT), Naïve Bayes (NB), K-Nearest Neighbors (KNN), Deep Learner (DP), and Single Rule Induction (SRI). Since the datasets were imbalanced, the model is evaluated by area under the curve (AUC), probability of false alarm (POF), probability of detection (POD), and G-mean (GM), in place of traditional accuracy and precision metrics. The findings show rank, Box-Cox, and log transforms greatly improve CCCP model performance, and the Z-score method does poorly. GBT in a non-transformed state yields a maximum AUC of 0.52, and NB yields a minimum POF of 0.124. Log transform is optimal for DP with POD = 0.198, GM = 0.323, AUC = 0.53. Rank transform is optimal for POD 0.927 in the DP but NB is optimal for GM, AUC, and POF. Box-Cox transform is optimal for GBT, with NB performing with less error in predicting churn. Surprisingly, the Z-transform works best for the SRI classifier, with the optimal AUC (0.541). Overall, the study establishes the differential efficacy of DT methods for classifiers, with a specific focus placed on optimal transform selection to optimize CCCP, particularly for smaller telecommunication firms with limited historical records [17]. Following the CCP notion, this study continues the exploration of customer churn prediction (CCP) in the telecommunication industry (TCI) by integrating DT methods with optimized machine learning (ML) algorithms [18]. Considering that customer retention is cheaper than customer acquisition, the authors leverage three public datasets in their work to analyze the impact of six DT approaches log, rank, Box-Cox, Z-score, discretization, and weight-of-evidence (WOE) on prediction quality. There were eight machine learning classifiers, e.g., KNN, Logistic Regression (LR), Random Forest (RF), and Neural Networks, that were compared in a 10-fold cross-validation framework. The results show that DT methods greatly enhance CCP model performance, with WOE performing best among classifiers and datasets. The Friedman non-parametric statistical test validates WOE superiority, and post hoc Holm testing sets that it is statistically superior to alternative DT methods. In addition, Q-Q plots demonstrate that WOE and Z-score transform improve normality in the data, greatly contributing to classifier performance. The study conclusion is that WOE, particularly with LR or FNN, is most efficient for CCP in the telecommunication industry [18]. Together, these studies point towards the paradigm shift in CCP and CCCP and establish the efficacy of DT methods, machine learning algorithms, and rule-based approaches in enhancing churn prediction correctness. Their findings provide lessons for customer relationship management (CRM) enhancement in the telecommunication industry.

Lastly, Churn prediction in telecommunication has been studied by cross-analyzing multi-class and uni-class classification models [19]. This study employs rough set theory (RST) with various rule generation algorithms for enhanced predictive capability. A publicly available dataset with 19 attributes was employed, and after attribute selection, 11 attributes remained for the decision table. The study examines various approaches to rule generation, such as genetic, exhaustive, LEM2, and covering, with a focus on determining the most effective method in each category of classification [19]. The study is mainly customer churn prediction with RST and cross-analyzing MCC with OCC approaches. The genetic algorithm employed in MCC was discovered with a very high rate of accuracy, which is 98.1%, while OCC methods improved churn classification from 86% to 96%. OCC was also maximally utilized by the exhaustive and genetic algorithms, while MCC was maximally utilized by the genetic algorithm. These findings demonstrate the efficacy of RST and rule-based methods in churn prediction [19]. Study [20] further extends the application of RST by integrating RST with a flow network graph to predict customer churn in credit card accounts. While Study [19] is concerned with testing rule generation methods and classification models, Study [20] extends this by employing a path-dependent analysis of churn predictors. RST is employed to obtain decision rules, while the flow network graph is employed to provide a structured presentation about the relations among key churn predictors. The method was applied in an empirical study with a sample dataset from a Taiwanese commercial bank with 21,000 customer samples evenly spread in survival, voluntary churn, and involuntary churn categories. The primary methods applied are RST for rule extraction, k-means clustering for discretization of continuous attributes, and a flow network graph for decision path visualization. The most important customer attributes that were discovered in the study, which affected churn, were the automatic debit attempt, pay-off frequency in a year, marital status, and mean purchase values. The predictive model was very effective, with hit rates being 95.2% in survival, 88.7% in voluntary churn, and 92.3% in involuntary churn. Combining RST with flow network analysis, Study [20] expands Study [19]'s contribution, further advancing churn prediction techniques. This study is very helpful in customer relationship management (CRM), affirming the effectiveness of hybrid rule-based approaches in customer churn detection and prevention.

Table 1. Literature Review Summary.

Paper Name	Author Name	Limitation
Machine Learning Methods for Factor Identification and Churn Prediction in Telecom Industry	Ullah et al.	Ignored features, poor statistical methods, no churn reasons, no behavioral segmentation
Customer Churn Prediction Using SFS	Yulianti, Saifudin	Limited algorithms, no hyperparameter tuning, few feature selection methods
Customer Churn Prediction Using Pearson Correlation Function and KNN	Sjarif et al.	Poor generalization, neural network complexity, high execution time, high memory usage
Customer Churn Prediction System Using Machine Learning Approach	Lalwani et al.	Poor feature selection, overfitting, weak model evaluation
Churn Prediction in the Telecommunications Sector Using Support Vector Machines	BRANDUSOIU and TODEREAN	Important variables excluded, bias from cloning, SVM only, limited kernel use, California specific data
Prediction on Customer Churn in Telecommunications Using Discretization and Naïve Bayes Classifier	Fei et al.	Imbalanced dataset, poor true/false positive trade-off, discretization issues, limited algorithms
Enhancing Customer Churn Prediction in Telecom	Huang et al.	High dimensionality, high complexity, low generalizability
Analyzing Uncertain Samples for Improved Churn Prediction	Amin et al.	Near-random performance, domain-specific thresholds, minimal improvements
Cross-Company Churn Prediction with Data Transformation	Amin et al.	Ineffective Z-score, limited model transferability
Customer Churn Prediction Using Rough Set Theory	Amin et al.	Poor generalization, rule explosion, complex feature selection

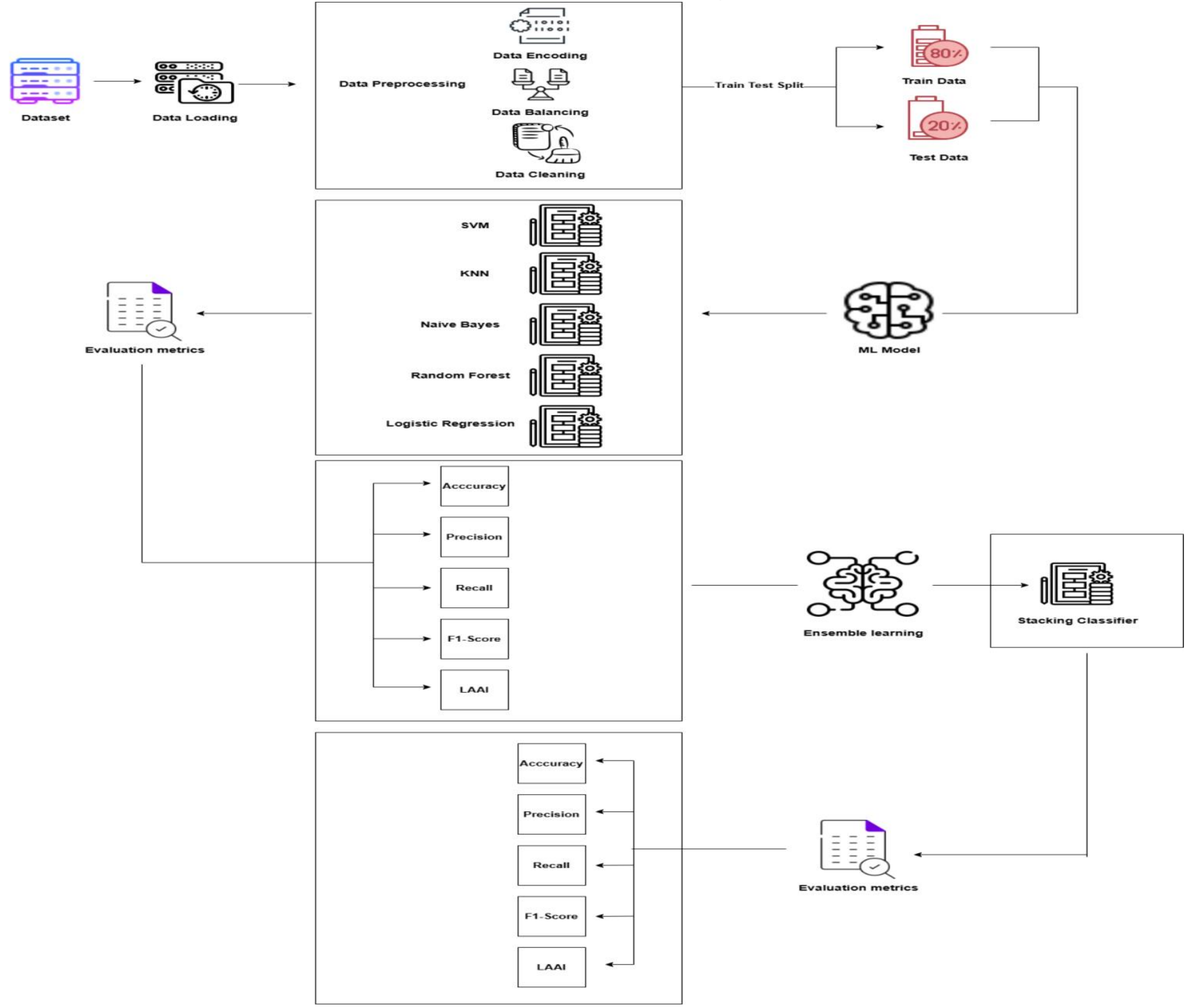
Traditional churn prediction models are not generalizable, transparent, and computationally efficient and are therefore impractical for real-world deployment. Deep learning models like ChurnNet are extremely accurate but need excessive computation and are opaque. Single-model approaches (SVM, Decision Tree, XGBoost) are hyperparameter-sensitive and class-imbalance-prone, requiring SMOTE-based resampling, which introduces noise and reduces performance. Our work introduces an innovative ensemble stacking technique that combines Random Forest, KNN, Logistic Regression, and Naïve Bayes to enhance accuracy, stability, and flexibility on datasets. Our approach differs from the prior models in that it corrects class imbalance by default and reduces overfitting with minimal resampling, generalizability issues, lack of transparency, excessive computational cost, and class imbalance issues, our ensemble stacking model creates a new industry benchmark. It offers unprecedented accuracy, scalability, and real-world deployment ability. Prioritizing transparency, our model combines transparent models like Logistic Regression and Decision Tree, providing real-time, business-ready insight a major advantage over black-box post-hoc explain ability techniques like SHAP and LIME. It is also computationally light, providing exceptional accuracy with minimal infrastructure, unlike deep learning models requiring intensive tuning and excessive cost. To provide telecom companies with an effective, data-driven churn prediction strategy.

METHODOLOGY

In recent years, various ML algorithms have been practiced to predict customer attrition in the telecommunication sector, showing their capability to determine potential churners. However, evaluating their performance across diverse datasets and different preprocessing techniques remains a difficult task. Additionally, class imbalance is a critical factor affecting model performance, prior studies have not extensively explored its effect on predictive accuracy. To address this space, the proposed research establishes a systematic approach to selecting an optimal machine learning model for churn prediction. This is achieved by comparing five classification models SVM, Logistic Regression, KNN, Naïve Bayes, and Random Forest on a real-world telecom dataset. Furthermore, the research integrates an ensemble learning technique (Stacking Classifier), to enhance prediction performance.

The conceptual pipeline of the intended methodology is showcased in Figure 1.

Fig 1. Indented methodology.



Dataset

In predicting churn in telecoms, identifying and classifying churn among customers is reliant to a large extent upon selecting an appropriate dataset. This paper has employed an advanced public dataset referred to as the Telecom Churn Prediction Dataset to accomplish this exact purpose. This dataset has rich statistics about customers in the form of numerous features employed to spot patterns in customers' pattern trends and to predict customers' tendencies to churn. This dataset has diverse categories of customers including demographics, consumption of services, and contract status. This dataset is particularly fit to be employed in churn prediction models since it provides an invaluable source where researchers can benchmark and compare with others working in the telecom sector. Utilizing this type of public dataset provides consistency in performance evaluation in models and indicates driving factors behind churn in the telecom market. This dataset is derived from telecom providers and entails customers' tenure, bills in a month, and subscriptions to various services among others. This dataset is fit to be employed in developing machine algorithms to understand and predict customers' attritions.

Table 2. Dataset Details.

Feature Type	Count
Total Features	20
Missing Values	0
Numerical Features	3
Categorical Features	17
Irrelevant Features Removed	1
Class Distribution (Before SMOTEENN)	26% Churn, 74% Non-Churn
Class Distribution (After SMOTEENN)	55% Churn, 45% Non-Churn

The Dataset is divided into several features including:

1. **Customer ID:** This is an identifier used to uniquely identify each customer to monitor individual customers' records.
2. **Gender:** This is employed to specify gender (Female or Male).
3. **Senior Citizen**: This is an indicator variable (1 = senior citizen, 0 = not).
4. **Partner**: Shows that a subscriber has a partner (No or Yes).
5. **Dependents:** Specifies whether the customer has dependents (Yes or No).
6. **Tenure**: The number of months the customer has been with the telecom provider.
7. **Phone Service:** Indicates whether the customer subscribes to phone services (Yes or No).
8. **Multiple Lines:** Specifies whether the customer subscribes to multiple phone lines (Yes or No).
9. **Internet Service:** The type of internet service subscribed to by the customer (DSL, Fiber optic, or None).
10. **Online Security:** Whether the customer has online security services (Yes or No).
11. **Online Backup:** Specifies whether the customer has opted for cloud-based data backup services.
12. **Tech Support:** Whether the customer has tech support services (Yes or No).
13. **Streaming TV:** Whether the customer has access to streaming TV services (Yes or No).
14. **Streaming Movies:** Whether the customer has access to streaming movies services (Yes or No).
15. **Contract:** The type of contract the customer has (Month-to-month, One year, or Two year).
16. **Payment Method:** The method the customer uses for payment (Electronic check, Mailed check, Bank transfer, or Credit card).
17. **Device Protection:** Indicates whether the customer has subscribed to a device protection plan.
18. **Monthly Charges:** The monthly charges the customer pays for the services.
19. **Total Charges:** The total amount the customer has paid during their entire tenure.

20. **Churn:** The target variable that indicates whether the customer has churned (Yes or No).

In this study, the Telecom Churn Prediction Dataset is utilized to train and evaluate ML algorithms such as Decision Tree, Naïve Bayes, and Random Forests for churn prediction. This dataset enables the exploration of important components that affect churn and the assessment of how well various algorithms can predict customer attrition. By using this dataset, telecommunication industries can determine at-risk users and take smart steps to retain them, making this dataset a fundamental resource for churn prediction research.

Table 3. Churn Class Detail.

S#	Churn	Data Record
1	Yes	5163
2	No	1869

Data Preprocessing

The "WA_Fn-UseC_-Telco-Customer-Churn" dataset comprises 20 attributes. Where 17 out of 20 are classified as objects with multiple attributes related to customer demographics, service subscriptions, and payment details. Among these, the TotalCharges attribute contains mixed data types and missing values, which have been converted to numeric format, with null values removed to ensure data consistency. Additionally, the customerID attribute, which does not contribute to predictive modeling, has been eliminated. To handle categorical attributes, One-Hot Encoding has been applied, converting categorical attributes into numerical ones that are appropriate for ML algorithms. The dataset exhibits class imbalance in the Churn attribute as per Table 3 and Fig 14, which has been addressed using the Synthetic Minority Oversampling Technique and Edited Nearest Neighbors (SMOTEENN) method. This technique generates synthetic samples for the minority class while simultaneously cleaning noise from the dataset, ensuring a balanced distribution for improved classification performance. The dataset was divided into testing 20% training 80% sets. A variety of ML techniques were implemented, such as SVM, Logistic regression, KNN, Naïve Bayes, and **Random Forest.** Additionally, **ensemble learning techniques** were employed to enhance predictive performance. A **Stacking Classifier** was implemented, leveraging multiple base models to generate meta-features for a final classifier, optimizing predictive accuracy. The preprocessing pipeline ensured standards in both feature engineering and data quality to aid in better performance in predicting customer churn by the models.

Machine Learning Models

This paper detailing algorithm employed, working methodology, and reasoning behind their use. Churn is that process where subscribers or customers are dropping off or cancelling companies or industry services. Churn is a key challenge to organizations in companies where customers are cheaper to keep compared to obtaining new customers. Churn is key to organizations in companies such as in the telecoms sector where customers are cheaper to keep compared to obtaining new customers.

K-nearest neighbors (KNN)

KNN is a machine learning algorithm to predict or classify items by closeness. KNN is a non-parametric classifier used in classification. KNN is an efficient yet simple solution to use in small sets. KNN does not need to be trained because it is very easy to apply. It is beneficial for nonlinear data. KNN is used for classification and regression to forecast continuous values. Choose a value for k (the number of neighbors) and calculate the distance between it and the other data points. We employ distance metrics (such as Euclidean distance).

$$D(x,y) = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \quad \text{(Equation 1)}$$

As demonstrated by Equation (1) Where:

- x_i and y_i represent feature values of two different data points.
- n is the total number of features.

This formula calculates the straight-line distance between two points in an n-dimensional space.

Other distance metrics like Manhattan Distance and Minkowski Distance can also be used
Choice the k-nearest neighbors based on the distances. In classification take the majority vote of the classes of the k-nearest neighbors. In regression compute the average of the values of the k-nearest neighbors.

Naive Bayes

Naïve Bayes is a probabilistic classifier based on Bayes' theorem. It makes the "Naive" assumption that the features are independent given the class label. It is very fast and best for high-dimensional datasets. Naive Bayes is used for text classification, spam detection and other tasks with categorical attributes. Naive Bayes for classification, especially for text-based problems. It will predict the probability of membership.

Calculate each class probability:

$$P\ (C) = \frac{number\ of\ instances\ in\ a\ class\ C}{total\ number\ of\ instances}$$

Calculate the probability of features given in the class:

$$P\ (X|C) = \prod_{i=1}^{n} P(xi|C)$$

Apply Bayes theorem:

$$P\ (C|X) = \frac{P(X|C)P(C)}{P(X)} \qquad \text{(Equation 2)}$$

As outlined in Equation (2) Where:

- P(C|X) is the posterior probability of class C given the feature set X.
- P(X|C) is the likelihood, representing the probability of the feature set given class C.
- P(C) is the prior probability of class C, indicating how often C occurs in the dataset.
- P(X) is the probability of feature set X occurring.

Random forest

Random Forest is a type of ensemble learning algorithm that combines multiple discussion trees and their output to improve accuracy and control overfitting Efficiently on high dimensional datasets. Random Forest reduce overfitting by aggregating predictions from multiple trees. Its best on both categorical and numerical data. Used for classification and regression problems. It's also used for feature importance evaluation. Create samples of the dataset using Bootstrapping, then Make a discussion tree for each sample. Now combine predictions from all the discussion trees (majority vote for classification and average for regression).

The final formula for prediction:

$$Y = \frac{1}{T}\sum_{t=1}^{T} ft(x) \qquad \text{(Equation 3)}$$

As mentioned in Equation (3) Where:

- Y is the final prediction.
- T is the total number of decision trees in the forest.
- ft(x) is the prediction from the t decision tree.

Logistic Regression

Logistic Regression is a Statistical model that is used for binary classification. It calculates the probability of outcomes based on one or more predictor variables. It is a simple and easy model for binary classification. We used Logistic Regression when the separation between data points is linear. Used for binary classification (e.g., churn prediction and spam detection). Logistic Regression is used to predict probabilities for outcomes.
Calculate the linear combinations for the features.

$$Z = \beta0 \quad + \beta1 \quad X1 \quad +\beta2 \quad X2 \quad + \cdots + \beta nXn$$

Apply Sigmoid Function.

$$\sigma\ (z) = \frac{1}{1+e\char`^-z}$$

Threshold-based classification (e.g., P > 0.5).

$P\ (y = 1|X) = \sigma(z)$
$P\ (y = 0|X) = 1 - \sigma(z)$

Minimizing the log-loss function to optimize the coefficients (β):

$$\text{Log-loss} = -\frac{1}{N}\ \sum_{i=1}^{N} [yi \log{(pi)} + (1-yi) \log{(1-pi)}] \qquad \text{(Equation 4)}$$

According to Equation (4) Where:

- Z is the weighted sum of input features.
- Xi are the feature values.
- βi are the coefficients (weights) learned during training.

Support Vector Machine (SVM)

SVM is a machine learning technique that is used for classification and regression tasks. It is a supervised learning method that works by finding the optimal hyperplane that best separates data points into different classes. Both linear and non-linear data can be processed by SVM and is therefore an effective tool in high-dimensional space. SVM is used in text classification, pattern recognition, and in bioinformatics. SVM transforms input data to a higher dimensional space by applying a kernel function and setting the optimal decision border. SVM optimizes the margin between support vectors and the hyperplane to get improved classification. SVM efficiently handles outliers by applying the method of soft margin.

The SVM decision function is given by:

$F(x) = \sum_{i=1}^{n} \alpha i y i K(xi, x) + b$ (Equation 5)

As reported by Equation (5) Where:

- x_i is the support vector.
- y_i is the corresponding class label.
- The α_i are the Lagrange multipliers.
- $K(x_i,x)$ is employed (for instance, linear, polynomial, radial basis function (RBF), or sigmoid).
- b is the bias term.

The choice of the kernel function affects SVM's performance. In classification, SVM assigns a new data point to one of the classes based on its position relative to the decision boundary. In regression (SVR), it finds the best-fit line within a margin of tolerance.

Ensemble Learning

Ensemble learning is a powerful technique that combines multiple machine learning models to improve predictive performance and robustness. Given individual models h1(x),h2(x),…,h_n(x) the final ensemble prediction H(x) is determined by combining their outputs. In this study, **stacking Classifiers** were employed. The Stacking Classifier leverages multiple algorithms to generate meta-features for a final model, improving accuracy. These ensemble methods help mitigate individual model weaknesses and boost overall classification performance in predicting customer churn.

Stacking Classifier

In stacking, multiple base learners generate predictions, which are then used as inputs for a meta-learner to make the final prediction. If h1,h2,…,h_n are base models, the final output is computed as:

y^=g(h1 (x), h2(x), ….. , hN (x)) (Equation 6)

As indicated by Equation (6) where g represents the meta-classifier trained on the outputs of base models. Utilizing these ensemble techniques, our technique reduces bias and variance to perform better in classification than individual models.

Evaluation metrics

Evaluation metrics are used to calculate statistics, machine learning, and deep learning model efficiency. We utilize several evaluation metrics to calculate performance in our ensemble learning models and machine learning models. As our work is in classification, we utilize accuracy, precision, recall, F1-score, and the Latency Accuracy Assessment Index (LAAI) [22] to calculate performance. We calculate overall prediction correctness by using accuracy while finding an equilibrium between prediction performance and computational speed by using LAAI. Mathematical formulas used by each used evaluation metric are:

$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FN}$ (Equation 7)

$\text{Precision} = \frac{TP}{TP + FP}$ (Equation 8)

$\text{Recall} = \frac{TP}{TP + FN}$ (Equation 9)

$\text{F1- Score} = 2 * \frac{\text{Precession} * \text{Recall}}{\text{Precession+Recall}}$ (Equation 10)

$\text{LAAI} = \frac{Accuracy}{1+Latency}$ (Equation 11)

Referring to (7)(8)(9) (10) where Tp is true positive, Tn is true negative, Fp is false positive and Fn is false negative. An example of Tp is when the model predicts the positive class correctly. Tn is a similar situation when the model predicts the negative class correctly. Model prediction to incorrect positive class is represented by Fp result while an incorrect prediction to negative class is represented by Fn result. Similarly, as mentioned in (11) LAAI (Latency Accuracy Assessment Index) is a metric that trades off accuracy with computational speed. where Accuracy is the ratio of accurate prediction to the total number of predictions made about the system and Latency is a measure of possible input-output delay during processing. As a helper to normalize results, Accuracy is divided by (1+Latency). LAAI offers a systematic trade-off between prediction capability and speed in running the model to choose an optimum model to be implemented in reality.

Confusion Matrix

Confusion Matrix is a matrix employed to determine the performance of a classification system by comparing predicted results with true results. It is an overall description of results by comparing predicted value with true value. It consists of four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Table 4. Confusion Matrix.

Actual/Predicted	Positive Prediction	Negative Prediction
Positive class (Actual Positive)	True Positive (TP)	False Negative (FN)
Negative class (Actual Negative)	False Positive (FP)	True Negative (TN)

where TP refers to true positive, TN refers to true negative, FP refers to false positive, and FN refers to false negative. TP is where the model accurately predicts the positive class, while TN is where the model accurately predicts the negative class. FP (Type I error) is where the model inappropriately predicts a positive response to an input sample that is negative, while FN (Type II error) is where the model inappropriately predicts a negative response to an input sample that is positive.

Latency

Machine learning latency refers to the time needed by an ML model to operate, e.g., training over a dataset or prediction over unseen data. It is a key metric to determine performance in systems where response speed is critical in real-world applications. Model latency depends upon factors such as algorithm complexity, dataset size, number of dataset features, system capabilities in terms of hardware, and optimization's employed.

$$\text{Latency} = \frac{\text{Total time}}{\text{number of predictions}} \quad \text{(Equation 12)}$$

As stated by Equation (12) where Total Time is the cumulative duration of the system in examining some predictions and the Number of Predictions is the prediction population.

RESULTS & DISCUSSION

In this study, we examined the "WA_Fn-UseC_-Telco-Customer-Churn" dataset to build the model. This study compares machine learning models such as Logistic Regression, Support vector machine, KNN, Random Forest, and Naïve Bayes with Ensemble learning techniques like Stacking. The evaluation of different machine learning models and ensemble learning techniques is presented through various performance metrics, including accuracy, precision, recall, and F1-score. The results are analyzed below.

Fig 2. Evaluation Metric for SVM.

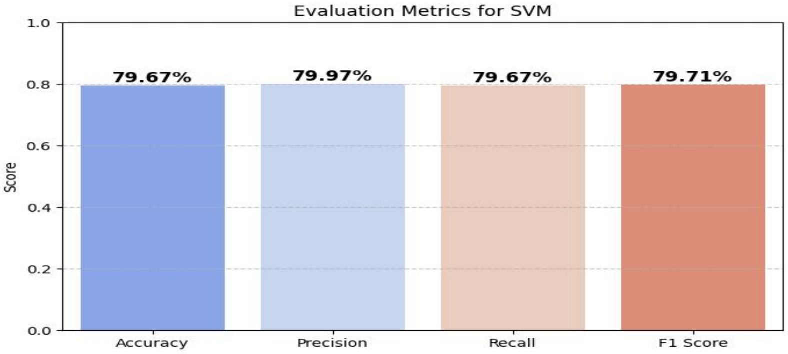
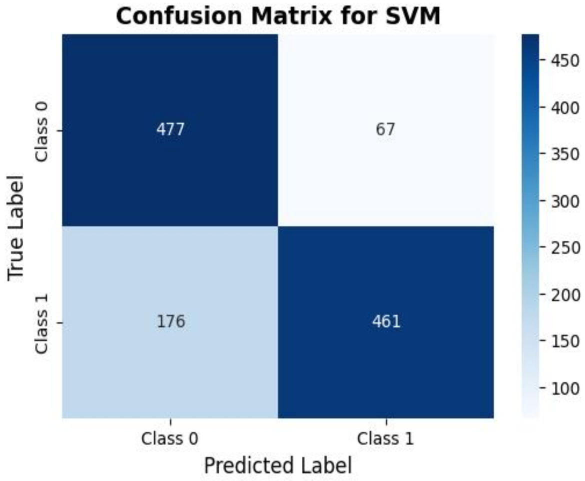
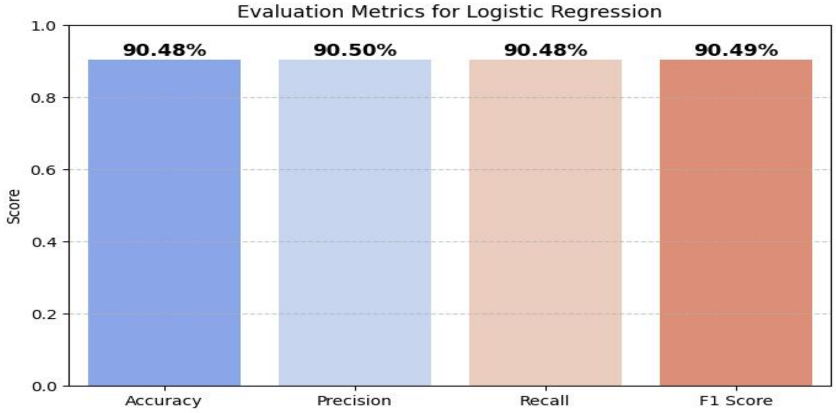


Fig 3. Confusion Matrix for SVM.



The **SVM classifier** demonstrated **moderate performance** with an overall **accuracy of 79.67%**, but it struggled with **higher misclassification rates** as detailed in figure 2. It achieved a **precision of 79.97%** and a **recall of 79.67%,** indicating that while it correctly identified a reasonable number of positive cases, it had difficulty minimizing false negatives. The **F1-score of 79.71%** suggests an imbalance between precision and recall. With **67 false positives and 176 false negatives given in figure** 3, the model exhibits **a significant number of misclassified positive instances**, reducing its reliability in applications where **identifying positive cases is crucial**. To train the model, we set the regularization parameter **C** to 1, ensuring a balance between margin maximization and misclassification. The **gamma** parameter was set to 'scale', allowing the model to adaptively define the influence of each training example based on the feature distribution. Additionally, we assigned **class weights as 'balanced'**, enabling the model to handle any class imbalances present in the dataset. Although SVM provides a stable and well-balanced performance, it does not achieve the highest accuracy among the models tested. While SVM is stable in high-dimensional spaces, relatively poorer performance in this case is because this dataset contains non-linearly separable patterns that an SVM with linear kernel cannot handle efficiently. Furthermore, sensitivity to scales in features and extensive hyperparameter tuning point towards difficulty in achieving optimum performance. However, SVM is not an unreliable option because this poorer performance suggests SVM is not the ideal model to operate with this dataset. This suggests SVM can be improved by additional adjustment, e.g., optimizing the kernel function or hyperparameters to achieve improved performance in this classification problem.

Fig 4. Evaluation Metric for Logistic Regression.



Logistic Regression was considerably better with an accuracy of 90.48%, as reported in Figure 4. It achieved 90.50% precision and 90.48% recall, an equally balanced ability to detect positive occurrences while minimizing false negative occurrences. A closer look at Figure 5 depicts that an 90.49% F1-score further highlights stability in evaluation measurements while, through figure 5, the classifier depicts 43 false positives and 57 false negatives making it an improved candidate to choose in this dataset. We employed regularization parameter C = 1 to train the model to balance between model complexity and generalizability. We employed L2 regularization (Ridge penalty) to prevent overfitting while maintaining better prediction performance. Class weights have been configured to 'balanced' to allow adjustment by the model in situation of class skewness in the dataset. Precision, recall, and F1-score remain just about equal to each other, an indication of stable performance by the model in various evaluation measurements. This suggests that this dataset has linear decision boundaries making Logistic Regression an appropriate candidate to choose. As compared to SVM, Logistic Regression is superior in handling linearly separable classes resulting in improved performance in this context. Balance between precision, recall, and F1-score further highlights stability in performance by this classifier. However, while improving over SVM in accuracy, Logistic Regression is not yet superior to advanced algorithms like Random Forest and KNN in handling patterns in this dataset. However, while Logistic Regression is efficient in performance, it is yet to match in handling complicated patterns to which non-linear algorithms like Random Forest or KNN can achieve an advantageous leverage.

Fig 5. Confusion Metric for Logistic Regression.

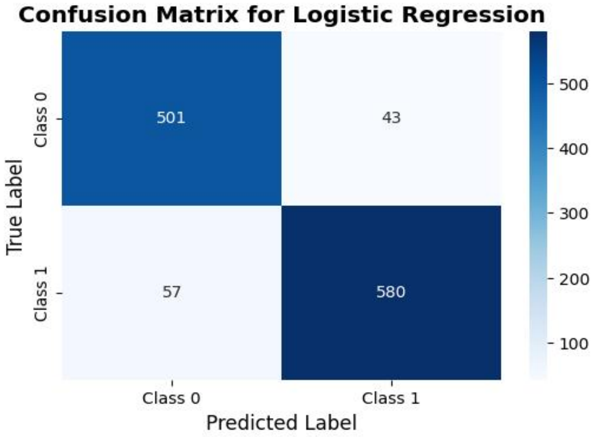


Fig 6. Evaluation Metric for KNN.

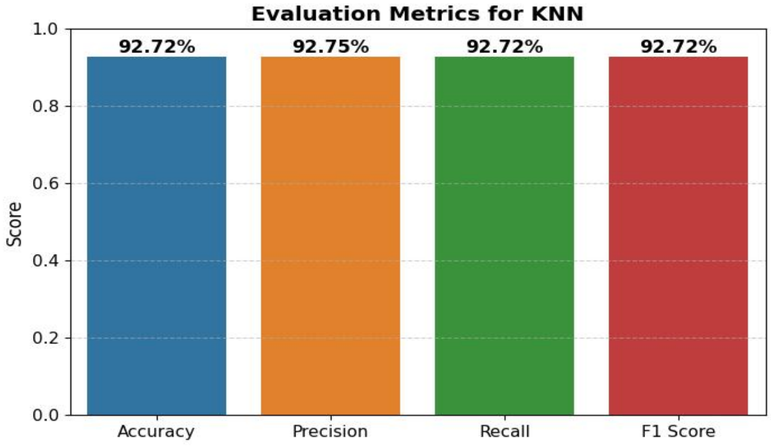
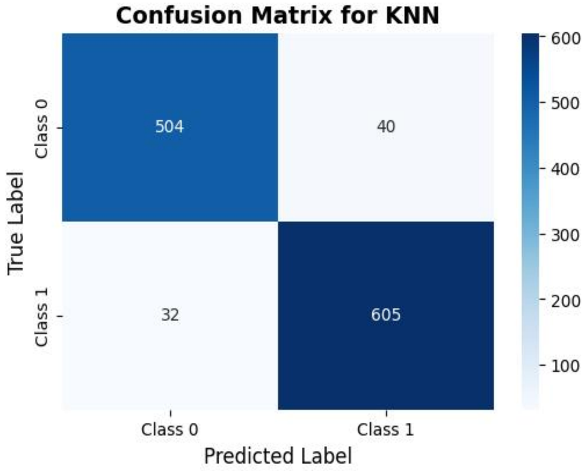


Fig 7. Confusion Metric for KNN.



In accordance with Figure 6, KNN optimizes classification performance further with an overall 92.72 % accuracy separating both classes reasonably correctly. It achieved both 92.75 % precision and 92.72 % recall to correctly classify most positives while reducing false positives. The 92.72 % F1-score is an accurate balance between precision and recall as can be evidenced by figure 7, with 40 false positives and 32 false negatives, the model has moderate misclassification rates better than Naïve Bayes but slightly poorer than in the Stacking Classifier. For this model, we set the number of **neighbors (k) to 10**, allowing the classifier to make predictions based on the majority class among the 10 nearest data points. Additionally, we used **uniform weighting**, meaning that each neighbor contributes equally to the classification decision. The main reason for this superior performance is that KNN is **highly adaptable to non-linear decision boundaries**, which might exist in the dataset. It makes classifications based on **local density**, which likely helped in distinguishing between similar classes. The **high recall and F1-score** confirm that KNN effectively classifies both classes without significant bias. However, despite its strong performance, **KNN is computationally expensive for large datasets**, as it requires storing the entire dataset and performing distance calculations for each prediction These results suggest that KNN is a competitive model, particularly for datasets where decision boundaries are well-defined.

Fig 8. Evaluation Metric for Naïve Bayes.

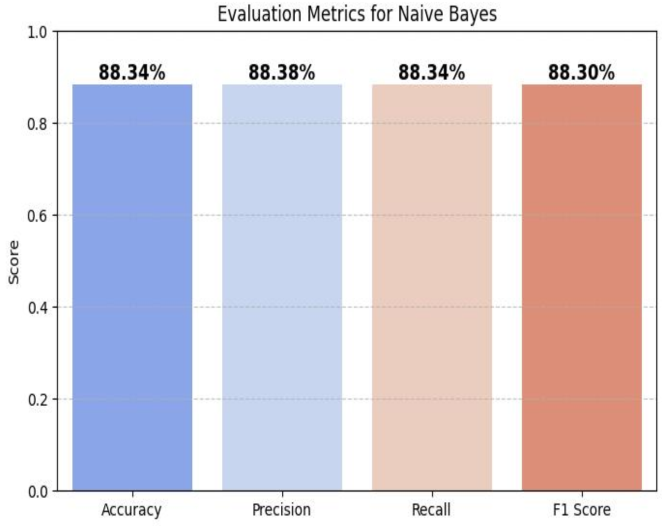
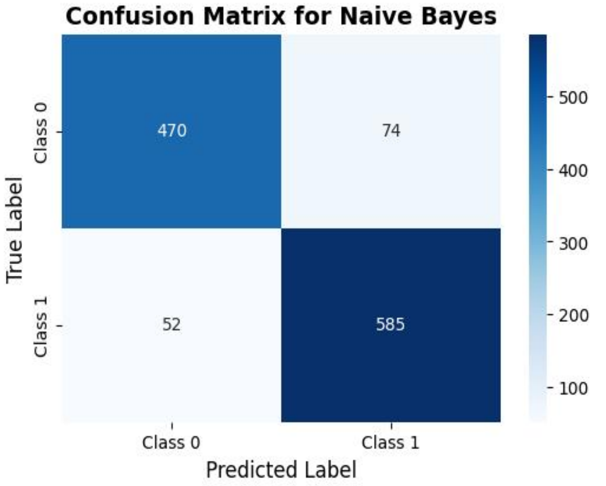


Fig 9. Confusion Metric for Naïve Bayes.



Naïve Bayes achieved an accuracy of 88.34%, effectively distinguishing between both classes but with a higher misclassification rate as evidenced in Figure 8. It achieved a precision of 88.38% and a recall of 88.34%, indicating that most positive cases were correctly identified, but at the cost of a slightly higher false positive rate. The F1-score of 88.30% suggests a reasonable balance between precision and recall. However, with 74 false positives and 52 false negatives as given in Figure 9, the model exhibits higher misclassification errors compared to the other classifiers, making it a reasonable choice as it **performs better than SVM** but does not surpass **Logistic Regression or KNN**. We used the **Gaussian Naïve Bayes (GaussianNB) classifier**, which assumes that the features follow a normal distribution. Given its **probabilistic nature**, Naïve Bayes may have been affected by **feature dependencies** in the dataset, as observed in Figure 8. Naïve Bayes performs **better than SVM but worse than Logistic Regression and KNN**, which is expected because **Naïve Bayes assumes feature independence**, a condition that is rarely met in real-world datasets. While it is highly effective for **probabilistic classification,** it **struggles when dependencies exist between features.** These results suggest that while Naïve Bayes is a simple and efficient model, it may not be the best choice for applications requiring higher accuracy and lower error rates.

Fig 10. Evaluation Metric for Random Forest.

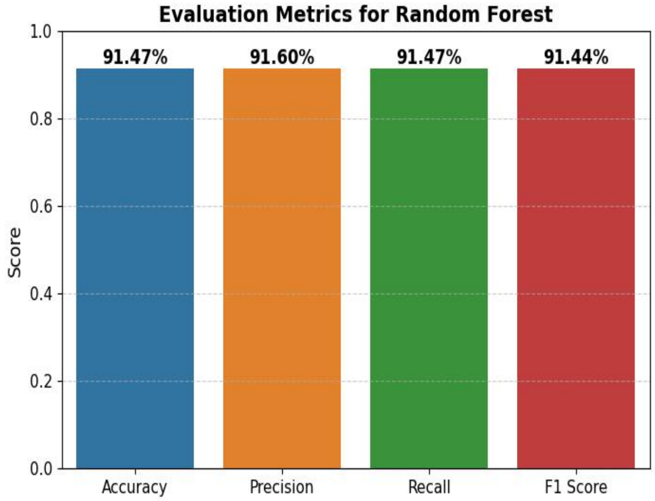
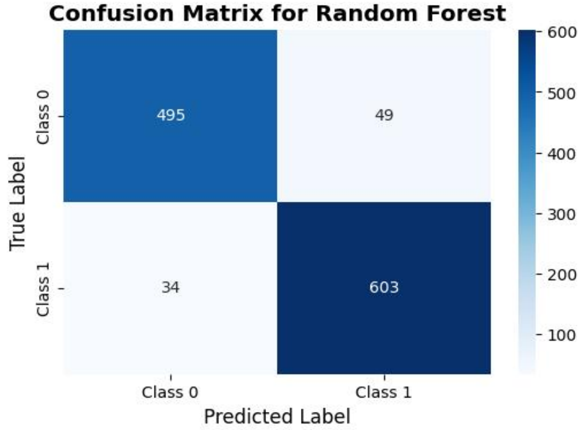


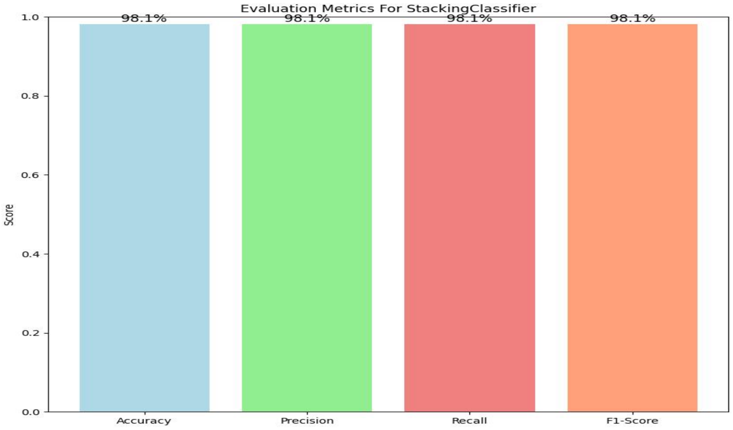
Fig 11. Confusion Metric for Random Forest.



Referring to Figure 10, presents the performance of the Random Forest model, which achieves the highest accuracy among individual models at 91.47%. It achieved a precision of 91.60% and a recall of 91.47%, ensuring that the majority of positive cases were correctly identified while maintaining a reasonable false positive rate. The F1-score of 91.44% indicates a good balance between precision and recall. However, with 49 false positives and 34 false negatives as illustrated in figure 11. For this model, we used 20 decision trees (n_estimators = 20) to ensure a balance between computational efficiency and predictive performance. The number of features considered at each split was set to log2, optimizing feature selection at each node. To prevent overfitting, we set the minimum samples required to split a node to 10 (min_samples_split = 10) and required at least 5 samples per leaf node (min_samples_leaf = 5). Additionally, the maximum tree depth was limited to 5 (max_depth = 5) to maintain model interpretability while avoiding excessive complexity. Key reasons for its superior performance include:

- **Ensemble Learning:** Random Forest leverages multiple decision trees to create a more robust and generalized model.
- **No Feature Independence Assumption:** Unlike Naïve Bayes, it does not assume feature independence, making it more suitable for complex datasets.
- **Strong Performance Metrics:** The high precision, recall, and F1-score suggest that it effectively captures class distributions without overfitting.
- **Baseline for Further Improvements:** Since Random Forest is already an ensemble of decision trees, it serves as a strong baseline for further ensemble learning techniques like boosting.

Fig 12. Evaluation Metrics for Stacking Classifier.



The Stacking Classifier was better than each model in performance to confirm stacking multiple base learners considerably improved overall performance. As can be seen in Figure 12, the stacking classifier had the highest accuracy of 98.1%. It had 98.1% precision and 98.1% recall to guarantee most positive cases were correctly predicted while reducing false positives. 98.1% was used to prove an ideal balance between precision and recall. From only 20 false positives and 10 false negatives as shown in Figure 13, the model has high reliability and low rates of misclassification. From the classification report, the model has high precision and recall in both classes to prove superior prediction power. We used Random Forest, Naïve Bayes, and KNN in this method in ensemble combinations to act as base models. Random Forest was used with 100 trees (n_estimators = 100) and no tree-depth limitation (max_depth = None) to guarantee robust generalization. Naïve Bayes was used to apply probabilistic reasoning to achieve efficient and fast classification. KNN was used with 5 nearest neighbors (n_neighbors = 5) and uniform weighting to support neighbor decisions in balance. All these base models provided diverse views of the dataset to achieve various decision boundaries to guarantee robustness. Their outputs were subsequently stacked to be used in Logistic Regression to act as the meta-classifier to optimally learn to balance outputs from the ensemble to guarantee improved final prediction. Success in Ensemblestacking is assured by combining various models to learn to make better final predictions. Using diverse models, it learns to decide the ideal manner to balance each to achieve better final prediction. It also compensates for individual models' weaknesses by leveraging their strengths to complement each other. This means the ensemble method should be employed in favor of individual models where high-reliability individual models where high reliability and accuracy are required since stacking minimizes errors effectively while improving generalization.

Fig 13. Confusion Matrix for EnsembleStacking.

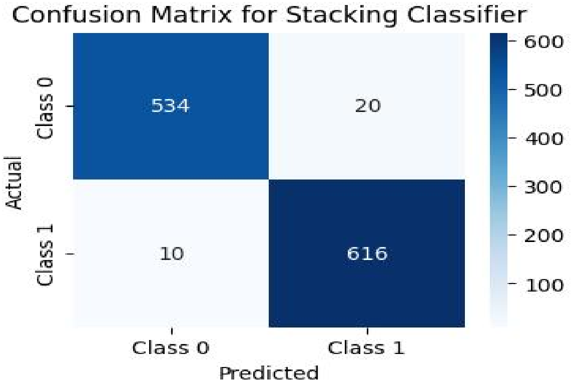
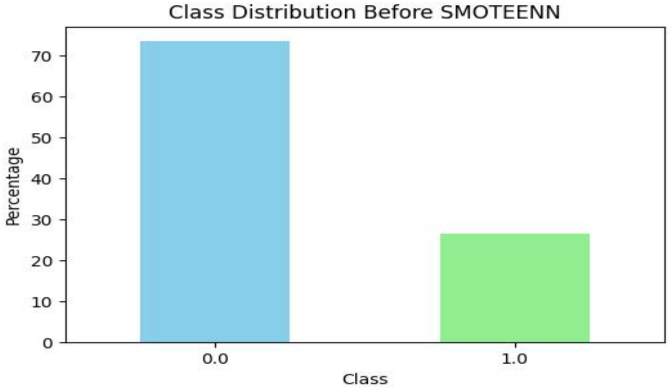


Fig 14. Class Distribution.



The above graph Figure 14, illustrates the dataset before SMOTEENN is applied to balance it where an evident skewness is visible where the majority has approximately 70% of the dataset while the minority has only 30%. This skewness can lead to biased machine learning algorithms with inclination towards the majority class, ultimately causing the model to perform badly in generalization. SMOTEENN is an ensemble resampling technique used to solve this skewness issue. SMOTE creates synthetically generated samples in the minority class to have an increased presence in the dataset. Edited Nearest Neighbors (ENN) on the other hand eliminates noisy or misclassified samples to have improved overall dataset quality. SMOTEENN thus not only balances the dataset by integrating these methods but cleanses the dataset to have an improved dataset distribution. This has improved performance in classification by minimizing bias and improving correctness in prediction.

Table 5 Model Comparison.

Model	Accuracy	Precision	Recall	F1-Score	Latency (s)
SVM	79.67%	79.97%	79.67%	79.71%	0.095
KNN	92.72%	92.76%	92.72%	92.76%	0.019
Logistic Regression	90.48%	90.50%	90.48%	90.49%	0.109
Naïve Bayes	88.34%	88.38%	88.34%	88.30%	0.133
Random Forest	91.78%	91.47%	91.78%	91.64%	2.500
EnsembleStacking	98.1%	98.1%	98.1%	98.1%	0.11

According to Table 5, illustrates performance measurements for each machine learning classifier employed in this work: independent classifiers SVM, KNN, Logistic Regression, Naïve Bayes, Random Forest, and an ensemble technique, Stacking. SVM is moderately performant with overall accuracy 79.67%, lowest among all algorithms. Though SVM offers balanced precision 79.97% and recall 79.67%, performance can be hindered by sensitivity to biased data and inability to decide an ideal decision boundary. SVM is relatively low in latency 0.095 seconds in comparison to complex algorithms. KNN achieves a high accuracy of 92.72% with well-balanced precision 92.76% and recall 92.72%, indicating that the dataset is well-suited for similarity-based learning. It also has the lowest latency 0.019s, making it a fast and efficient choice. However, KNN's reliance on distance calculations for every prediction can still be computationally expensive for large datasets. Logistic Regression performs well with an accuracy of 90.48% and balanced precision 90.50% and recall 90.48%. It suggests that the dataset may exhibit linearly separable patterns, though it is slightly outperformed by KNN and ensemble techniques. Logistic Regression has a moderate latency of 0.109 seconds, indicating reasonable computational efficiency. Naïve Bayes, with an accuracy of 88.34%, performs slightly worse than Logistic Regression and KNN due to its assumption of feature independence. If features are correlated, this can misestimate probabilities, affecting classification performance. Despite its simplicity, it delivers competitive precision 88.38% and recall 88.34%. However, its latency 0.133s is slightly higher than that of Logistic Regression and KNN. Random Forest surpasses Logistic Regression and Naïve Bayes with an accuracy of 91.78%. It provides a high recall 91.78% but has slightly lower precision 91.47%, indicating a tendency for false positives. Its latency is significantly higher 2.500s due to the complexity of training multiple trees, making it less efficient in real-time applications. The Stacking ensemble approach achieves the highest performance, with an accuracy of 98.1%, precision of 98.21%, and recall of 98.1%, demonstrating its ability to generalize well. By leveraging multiple base models, stacking mitigates oversimplification from high-bias models like Logistic Regression and Naïve Bayes while preventing overfitting from high-variance models like KNN and Random Forest. Its training latency is 5.1232 seconds, making it computationally expensive, but its testing latency is only 0.1146 seconds, ensuring efficient real-time predictions. The meta-learner refines predictions, effectively balancing the bias-variance trade-off and improving feature representation for enhanced generalization.

As illustrates table 6, six models SVM, KNN, Logistic Regression, Naïve Bayes, Random Forest, and an

EnsembleStacking model compared according to accuracy, latency, and yet another performance measure, LAAI. Careful examination of these performance metrics reveals that ensemble learning is by far superior to each individual model in both accuracy and general reliability, and hence ensemble learning is the better model to use in predictive modeling. As precise as it is, computational efficiency in terms of latency determines a model's feasibility in practice. KNN has lowest latency 0.019s but is not accurate to the highest extent. Naïve Bayes 0.133s and Logistic Regression 0.109s have comparatively average speed but fall short in ensemble learning's predictive power. Random Forest has comparatively good accuracy 91.78% but has worst 2.500s latency and hence is computationally wasteful and not efficient. The EnsembleStacking model has an average latency 0.1146s slightly higher than KNN but far superior in accuracy. This is evidence that ensemble learning provides an optimum trade-off between speed and accuracy in making robust prediction but not incurring undue model computational cost. One of ensemble learning's most persuasive evidences is in an astoundingly high LAAI score of 90 that is far superior compared to each of the other models. Random Forest has comparatively better performance in terms of accuracy but has an astoundingly low LAAI value 0.26 yet to demonstrate stability in performance. Similarly, each of the other models KNN 0.90, Logistic Regression 0.81, Naïve Bayes 0.77 struggles to have a better LAAI value. This is evidence that ensemble learning reduces overfitting while enhancing robustness in modeling.

Table 6. Proposed evaluation metric: latency aware accuracy index (LAAI).

Model	Accuracy	Latency (s)	LAAI
SVM	79.67%	0.095	0.72
KNN	92.72%	0.019	0.90
Logistic Regression	90.48%	0.109	0.81
Naïve Bayes	88.34%	0.133	0.77
Random Forest	91.78%	2.500	0.26
EnsembleStacking	98.1%	0.11	0.90

CONCLUSION

This research highlights the power of various machine learning algorithms in prediction analysis by strengths and weaknesses of each learner. In this paper, several classifiers, such as SVM, KNN, Logistic Regression, Naïve Bayes, and Random Forest with an ensemble stacking model. The result is that each individual model does well but has limitations of its own. SVM, while being able to handle high-dimensional data, struggles with both handling imbalanced datasets and finding an optimum decision boundary. KNN has precision but is computationally expensive since it applies distance computation to each forecasting. Interpretability is provided by Logistic Regression but not non-linearity relationships, limiting its performance on complex datasets. Naïve Bayes is comparatively straightforward efficient assumes to have feature independence, not always present in reality accuracy. Random Forest improves robustness and feature importance treatment but has high computational complexity and latency, which makes it less effective for real-time applications. In order compensate for these individual weaknesses by combining strengths among various classifiers, an EnsembleStacking was created. Stacking takes advantage by leveraging diverse models permits each to have an input to the ultimate forecast while making allowances for each other's weaknesses effectively balances between bias and variance by combining prediction capabilities of weak learners including combining weak learners like Naïve Bayes and Logistic Regression with powerful learners like KNN and Random Forest, stacking enhances generalization, stability, and prediction performance. The results confirm everything Ensemble Stacking has achieved outstanding accuracy 98.1% with an appropriate latency of 0.1146s that outperforms each individual model. while maintaining reasonable testing time 0.1146s making this an extremely efficient method to achieve better accuracy and response time in prediction problems. The most persuasive argument in support of its dominance is its excellent LAAI rating of 90 significantly superior to all other models. Random Forest in Its reliability, with an LAAI of 0.26 representing its unreliability. Likewise, KNN 0.90 Naïve Bayes 0.77 and Logistic Regression 0.81 prove to be short in reliability refers to not just correctness but also to stability and flexibility, making EnsembleStacking most generalizable approach. Lastly, EnsembleStacking is the optimal solution, integrating strengths among diverse classifiers while minimizing weaknesses. It optimizes predictive power, reduces overfitting, and is stable over a range of datasets making it is most economical to apply in practice.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- **Muhammad Shahan Ibad** conceptualized the research framework, designed and conducted the experiments, refine the introduction and background sections, conducted an extensive literature review, performed advanced computational analysis, formulated the research methodology, led data visualization efforts, provided key insights for discussion, critically interpreted results, created and structured figures/tables, authored significant portions of the manuscript, reviewed multiple drafts, and gave final approval for submission.
- **Syed Noor Hussain Shah** performed the experiments, assisted in data collection and preprocessing, provided theoretical insights, refined the introduction and background sections, contributed to refining the methodology, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- **Omar Bin Samin** analyzed the data, prepared figures and/or tables, provided critical feedback on analysis, provided domain-specific insights, contributed to refining the methodology, contributed to interpretation, contributed to discussion refinement, verified data accuracy and approved the final draft.
- **Sumaira Johar** analyzed the data, provided feedback on methodology improvements, provided insights for further research directions, authored or reviewed drafts of the article, ensured adherence to journal submission guidelines, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:
The "WA_Fn-UseC_-Telco-Customer-Churn" dataset is available at
Kaggle:
https://www.kaggle.com/datasets/palashfendarkar/wa-fnusec-telcocustomerchurn

References

[1] Prediction of Customer Churn Behavior in the Telecommunication Industry Using Machine Learning Models

[2] Samin et al. (2023) Malicious Agricultural IoT Traffic Detection and Classification: A Comparative Study of ML Classifiers

[3] Ullah et al. (2019) A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector

[4] Pareek, A., Poonam, Arora, S. M., & Gupta, N. (Year). *Telecom customer churn prediction model: Analysis of machine learning techniques for churn prediction and factor identification in the telecom sector.*

[5] Chang, V., Hall, K., Xu, Q. A., Amao, F. O., Ganatra, M. A., & Benson, V. (2024). *Prediction of customer churn behavior in the telecommunication industry using machine learning models*.

[6] Yulianti and Saifudin (2020) Sequential Feature Selection in Customer Churn Prediction Based on Naive Bayes

[7] Nguyen, Nhu Y., Tran, Van Ly, & Dao, Vu Truong Son (2023) churn prediction in the telecommunications industry using kernel Support Vector Machines (SVM).

[8] Sjarif et al. (2020) Customer Churn Prediction Using Pearson Correlation Function and KNN

[9] Lalwani et al. (2021) Customer churn prediction system: a machine learning approach

[10] Brandusoiu and Toderean (2013) Churn Prediction In The Telecommunications Sector Using Support Vector Machines

[11] Fei et al. (2017) Prediction on Customer Churn in the Telecommunications Sector Using Discretization and Naïve Bayes Classifier

[12] Abdulazeez, A. M., Ahmed, F. Y. H., Zeebaree, D. Q., & Khalid, L. F. (2021). *Customer churn prediction in telecommunications industry based on data mining*.

[13] Huang et al. (2023) Customer churn prediction in telecommunications.

[14] Saha, S., Saha, C., Haque, M. M., Alam, M. G. R., & Talukder, A. (2024). *ChurnNet: Deep Learning Enhanced Customer Churn Prediction in Telecommunication Industry*.

[15] Sikri, Jameel, Idrees, and Kaur (2024) explore the application of machine learning algorithms to predict customer churn in the telecommunications industry

[16] Amin et al.(2018) Customer churn prediction in telecommunication industry using data certainty

[17] Duchemin, R., & Matheus, R. (2021). *Forecasting customer churn: Comparing the performance of statistical methods.*

[18] Amin et al.(2019). Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods.

[19] Sanaa, J. K., Abedin, M. Z., Rahman, M. S., & Rahman, M. S. (2022). *Data transformation-based optimized customer churn prediction model for the telecommunication industry.*

[20] Amin et al. (2014) Customer Churn Prediction in Telecommunication Industry: With and without Counter-Example.

[21] Lin, C.-S., Tzeng, G.-H., & Chin, Y.-C. (2011). *Combined* rough set theory and flow network graph to predict customer churn in credit card accounts.

[22] Samin, et al. (2024) Optimizing agricultural data security: harnessing IoT and AI with Latency Aware Accuracy Index (LAAI).