Iranian Journal of Operations Research
Vol. 16, No. 2, 2025, pp. 156-188

An Automated Stacking Framework for Insurance Customer Profitability Prediction using Hybrid Transformer-Gradient Boosting Architectures

Amirhossein Malakouti Semnani[1], Sohrab Kordrostami[21], Amirhossein Refahi Sheikhani[3], Mohammad Hossein Moattar[4]

Insurance companies face the critical challenge of identifying "good customers"—policyholders who consistently pay premiums with minimal or no claims—within large, heterogeneous datasets. This study proposes and evaluates a hybrid machine learning framework to predict good customer status using an enhanced insurance dataset that integrates demographic, financial, and policy-related features. The framework combines an XGBoost classifier, a soft-voting ensemble of RandomForest and LightGBM, and a custom Transformer Encoder, with all models tuned using the Optuna hyperparameter optimization library to enhance predictive accuracy and interpretability.

The methodology includes preprocessing steps such as categorical encoding and standardization of numerical variables (e.g., age, BMI, premium with GST), followed by a novel label engineering scheme that defines good customers as those whose premiums exceed the mean plus one standard deviation and have no claim history. The dataset is split into training (80%) and testing (20%) subsets. Two hybrid architectures are developed: Model A, which fuses the predicted probabilities from XGBoost and the Transformer Encoder using a 60–40 weighting, and Model B, which employs a soft-voting ensemble of RandomForest and LightGBM. Ablation studies quantify the contribution of each component, while performance is assessed using accuracy, AUC, F1-score, and Matthews Correlation Coefficient, supported by visual tools such as correlation heatmaps, ROC curves, and confusion matrices.

Experimental results show that Model A attains an accuracy of 0.8720 and an AUC of 0.9140, whereas Model B achieves an accuracy of 0.8850 and an AUC of 0.9260 after systematic hyperparameter tuning. Removing either the Transformer or XGBoost markedly degrades Model A, while omitting RandomForest or LightGBM leads to smaller performance drops in Model B, underscoring the value of ensemble diversity. Overall, the proposed framework provides a practical tool for insurance customer segmentation and profitability-oriented decision-making, and its open-source implementation facilitates replication, extension with additional features or larger datasets, and potential real-time deployment in operational insurance environments.

***Keywords**: Insurance Customer Profitability Prediction, Hybrid Transformer-Gradient Boosting Architectures, Automated Stacking Framework.*

Corresponding author,
[1] Department of Mathematics and Computer Sciences, La.C., Islamic Azad University, Lahijan, Iran, Email: a.malakoutisemnani@iau.ac.ir.
[2] Department of Mathematics and Computer Sciences, La.C., Islamic Azad University, Lahijan, Iran, Email: sohrab.kordrostami@iau.ac.ir.
[3] Department of Mathematics and Computer Sciences, La.C., Islamic Azad University, Lahijan, Iran, Email: ah.refahi@iau.ac.ir.
[4] Department of Computer Engineering, Ma.C., Islamic Azad University, Mashhad, Iran mhmoattar@iau.ac.ir.

1. Introduction

The insurance industry plays a central role in modern financial systems by providing risk transfer mechanisms for individuals and organizations. In recent years, rapid digitalization and the availability of large-scale customer data have fundamentally transformed insurance operations, enabling a shift from traditional actuarial methods toward data-driven decision-making. While this transition creates significant opportunities for improved pricing, underwriting, and customer segmentation, it also introduces challenges related to the analysis of high-dimensional, heterogeneous datasets that combine demographic, financial, and behavioral information.

A key strategic objective for insurance companies is the identification of so-called *good customers*—policyholders who generate stable revenue through relatively high premium payments while exhibiting low claim frequency. Accurately identifying such customers can improve portfolio profitability, reduce underwriting risk, and support targeted retention and marketing strategies. However, conventional statistical approaches, including generalized linear models and rule-based actuarial techniques, often struggle to capture the complex, non-linear interactions among variables such as age, income, health indicators, credit score, and claim history. As a result, their predictive power is limited when applied to modern insurance datasets characterized by rich feature interactions and evolving customer behavior.

Machine learning (ML) models have therefore become increasingly prominent in insurance analytics. Ensemble-based methods, particularly gradient boosting frameworks such as XGBoost, have demonstrated strong performance in tasks including claim prediction, fraud detection, and premium estimation by effectively modeling non-linear relationships and feature interactions [6], [8], [11]. Despite their advantages, tree-based ensembles can be sensitive to hyperparameter selection and may exhibit reduced generalization performance when not carefully tuned. In parallel, recent advances in deep learning have introduced transformer-based architectures for tabular data, extending self-attention mechanisms originally developed for natural language processing to structured datasets. Models such as Tab Transformer and FT-Transformer have shown promising results by explicitly learning feature-wise dependencies, although they often require greater computational resources and larger datasets to achieve consistent gains over tree-based methods [10], [8].

Given these complementary strengths and limitations, hybrid modeling strategies that combine tree-based ensembles with transformer-based architectures represent a promising but relatively underexplored direction in insurance analytics. Existing studies typically evaluate these models in isolation, with limited attention to profitability-oriented label design, systematic hyperparameter optimization, or rigorous ablation analysis. Moreover, many prior works focus on claims, fraud, or churn prediction, rather than explicitly defining customer value in terms of both revenue generation and risk exposure.

To address these gaps, this study proposes a hybrid machine learning framework for predicting high-value insurance customers using an enhanced insurance dataset that integrates demographic, financial, and policy-related features. A profitability-oriented definition of a good customer is adopted, identifying policyholders whose premium payments exceed the mean plus one standard deviation while exhibiting no claim history. Two hybrid architectures are developed. Model A combines a custom Transformer Encoder with an XGBoost classifier through weighted probability fusion, while Model B employs a soft-voting ensemble of Random Forest and LightGBM. All models are systematically tuned using the Optuna hyperparameter optimization framework to ensure robust performance and fair comparison [1].

The main contributions of this work are threefold. First, it introduces a domain-specific, profitability-driven label engineering strategy for identifying valuable insurance customers. Second, it proposes and empirically evaluates two hybrid learning architectures that integrate transformer-based deep learning with ensemble tree methods, supported by detailed ablation studies. Third, it provides a reproducible and extensible experimental framework that can be applied to larger or real-world insurance datasets and potentially deployed in operational decision-making environments.

The remainder of this paper is organized as follows. Section 2 reviews related work and highlights existing research gaps. Section 3 describes the dataset, preprocessing pipeline, and individual learning models. Section 4 details the proposed hybrid methodologies and evaluation strategy. Section 5 presents the experimental results and performance analysis, followed by a discussion of implications in Section 6. Finally, Section 7 concludes the paper and outlines directions for future research.

2. Previous Studies

Identifying high-value customers in the insurance industry—policyholders who consistently generate revenue through premium payments while exhibiting limited claim activity—has been widely studied in the context of risk assessment, pricing, and profitability analysis. As insurance datasets have grown in scale and complexity, traditional statistical approaches have increasingly been supplemented or replaced by machine learning (ML) techniques capable of modeling non-linear relationships and high-dimensional feature interactions. Existing research related to this problem can be broadly categorized into six main themes: machine learning models for insurance analytics, ensemble and voting-based approaches, hyperparameter optimization, transformer-based models for tabular data, techniques for handling class imbalance, and interpretability with business deployment considerations.

2.1. Machine Learning and ensembles in insurance

Gradient boosting algorithms, particularly XGBoost, have become prominent tools in insurance prediction tasks due to their strong performance on structured data. Chen and Guestrin demonstrated that XGBoost achieves superior AUC and normalized Gini scores compared with AdaBoost and neural networks in large-scale classification problems [6]. Subsequent studies have confirmed these findings in insurance-specific contexts. For example, Nyström and Witt reported substantial improvements in vehicle premium prediction accuracy when using XGBoost instead of linear regression and multilayer perceptrons on real insurance datasets [18]. In fraud detection scenarios, Averro et al. showed that imbalance-aware variants of XGBoost can significantly improve minority-class recall, emphasizing the importance of loss design and class weighting in insurance applications characterized by skewed class distributions [17].

Beyond individual learners, ensemble methods have been extensively explored to improve robustness and generalization. Njoh-Paul compared stacking and boosting approaches with individual classifiers such as logistic regression, SVM, and neural networks, concluding that ensemble strategies offer a more balanced trade-off between accuracy, sensitivity, and AUC [17]. Similarly, Lin et al. proposed a RandomForest-based ensemble framework for insurance big data, demonstrating improvements in both predictive accuracy and computational efficiency over traditional classifiers [12]. More recent work by Gutiérrez-Gallego et al. introduced balanced under-bagged ensembles for imbalanced motor insurance datasets, achieving improved recall without sacrificing overall

performance [9]. These studies collectively highlight the effectiveness of ensemble diversity in addressing the complexity of insurance data.

2.2. Hyperparameter optimization and transformers for tabular data

As model complexity has increased, hyperparameter optimization has become a critical component of ML pipelines in insurance analytics. Akiba et al. introduced Optuna, a Bayesian optimization framework based on Tree-structured Parzen Estimators, demonstrating its efficiency in discovering high-performing configurations for both gradient boosting and deep learning models. Subsequent studies in domains such as healthcare and financial risk prediction have shown that Optuna-tuned models often outperform those optimized via grid or random search, particularly in imbalanced classification settings [1].

In parallel, transformer architectures originally developed for natural language processing have been adapted to tabular data. TabTransformer applies self-attention to contextualized categorical embeddings and has demonstrated consistent performance gains over gradient-boosted trees across multiple benchmark datasets [10]. Gorishniy et al. further refined this idea with the FT-Transformer, showing that carefully regularized transformer encoders can match or surpass tree-based ensembles under certain conditions [8]. However, the practical applicability of tabular transformers remains debated. Badaro et al. noted that such models often require larger datasets and incur higher computational costs, which may limit their adoption in production insurance systems compared with more efficient tree-based methods [10].

2.3. Imbalanced data, interpretability, and business impact

Class imbalance is pervasive in insurance problems such as fraud detection and high-value customer identification, where positive cases may constitute less than 5% of the portfolio. Matharaarachchi et al. proposed a SMOTE variant that down-weights Local Outlier Factor outliers, improving F1-score and PR-AUC across a large collection of imbalanced datasets [14]. Fernanda et al. compared random oversampling, SMOTE, and SOMO for vehicle insurance fraud and recommended SMOTE-based strategies for achieving robust performance across multiple classifier families. Nonetheless, several studies indicate that strong learners such as XGBoost and LightGBM can sometimes reach comparable performance through probability-threshold adjustment or class-weight tuning without explicit oversampling [7].

With rising model complexity, explainable AI (XAI) tools have become essential. Lundberg and Lee introduced SHAP as a unified framework for feature attribution [13], and subsequent work by Tabari et al. showed that SHAP-based importance yields clinically meaningful explanations in radiomics and can be extended to structured risk models [21]. In insurance, Pinnacle Actuaries proposed a multiplicative SHAP formulation aligned with actuarial rate relativities, enabling transparent decomposition of premium differences driven by variables such as claim history and credit score [19]. On the business side, Spedicato et al. demonstrated that gradient boosting for pricing can significantly increase margin gain relative to generalized linear models [20], and McKinsey reported sizable improvements in pricing accuracy and underwriting speed from ML-driven workflows [15].

2.4. Research gaps and position of this study

Despite extensive work on ML for insurance, several gaps remain. Most studies focus on claims, fraud, or churn, and rarely adopt an explicit profitability-oriented definition of a "good customer" combining high premium with a claim-free record. While transformer models and tree-based ensembles have been studied separately, there is little research on hybrid architectures that fuse transformer encoders with XGBoost via weighted probability combination or combine Random Forest and LightGBM in soft-voting ensembles tailored to customer profitability. Systematic ablation studies examining the marginal contribution of each ensemble component are uncommon, and many experiments rely on synthetic or Kaggle-style datasets with limited discussion of cost-sensitive or profit-based evaluation metrics.

The present study addresses these gaps by (1) defining good customers as high-premium, claim-free policyholders, (2) proposing two hybrid architectures—Model A (Transformer + XGBoost with weighted fusion) and Model B (RandomForest + LightGBM with soft voting), (3) applying Optuna-based hyperparameter optimization with ablation analysis to quantify component contributions, and (4) providing an open-source, reproducible framework oriented toward insurance customer profitability and suitable for future deployment on larger or real-world datasets.

3. Materials and Methods

This section describes the dataset, computational environment, and the machine learning frameworks used in the proposed hybrid models for predicting good customer status in insurance. The three core models—XGBoost, a VotingClassifier combining RandomForest and LightGBM, and a custom Transformer Encoder—are presented separately to clarify their configurations and roles. The overall architecture of the proposed framework is illustrated in Figure 1, depicting the transition from data preprocessing and Optuna-based hyperparameter optimization to final model fusion and evaluation.

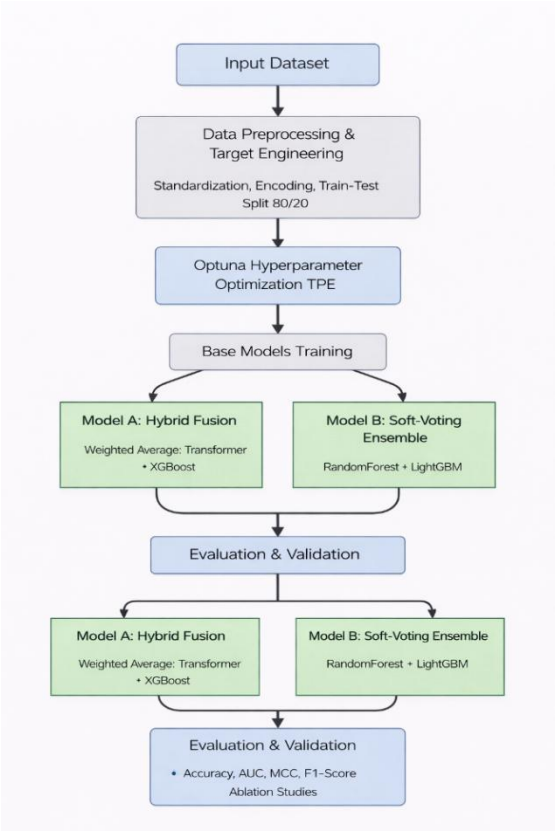

Figure 1: Overview of the proposed hybrid machine learning framework for customer classification.

3.1. Dataset Description

The primary dataset, enhanced_insurance_data, sourced from Kaggle, is a simulated insurance dataset comprising approximately 10,000 records and 25 features grouped into demographic, financial, and policy-related categories. Demographic variables include Age, Gender, Smoker, Marital_Status, Number_of_Dependents, and Occupation; financial attributes cover Annual_Income, Premium_with_GST, and Sum_Insured; and policy-related features consist of Claim_History, Policy_Period, Health_Score, Credit_Score, Previous_Claims, and Risk_Score. The dataset was validated for consistency and integrity, with missing values handled implicitly by algorithms that support native missing-value treatment, such as XGBoost, and explicitly during preprocessing for other models using mean imputation for numerical features and mode imputation for categorical variables [6].

The binary target variable Good_Customer was engineered using a profitability-oriented criterion: a customer is labeled as good if their premium payment exceeds the mean premium plus one standard deviation and their claim history is zero $(\text{Claim_History} = 0)$, following the approach of Nyström and Witt [18]. This definition identifies customers who contribute higher revenue while exhibiting minimal risk and yields a class distribution of roughly 30% good customers and 70% non-good customers, indicating a moderately imbalanced dataset suitable for supervised learning.

This profitability-oriented criterion is formally defined as follows:

$$Y_i = 1\ if\left(Premium_i > \mu_{premium} + \sigma_{premium}\right) AND(Claims_i = 0); 0\ otherwise \quad (1)$$

where Y_i is the binary label for customer i, Premium_i is the premium payment with GST, $\mu_{premium}$ and $\sigma_{premium}$ are the mean and standard deviation of the premium distribution, and Claims_i is the claim history flag ($0 = \text{no claims}$).

3.2. Computational Environment

Computational experiments were conducted on a Windows 10 Professional 64-bit workstation equipped with an Intel Core i7-10700 CPU (8 cores, 2.9 GHz base frequency), 16 GB DDR4 RAM, and an NVIDIA GeForce RTX 3060 GPU with 12 GB VRAM, which was used to accelerate Transformer Encoder training. The Python 3.9 environment, managed via Anaconda, included key libraries for data manipulation (pandas 1.4.2, numpy 1.22.0), machine learning (scikit-learn 1.0.2, xgboost 1.5.0, lightgbm 3.3.1), deep learning (torch 1.9.0 with CUDA 11.1), hyperparameter optimization (optuna 2.10.0), visualization (matplotlib 3.5.1, seaborn 0.11.2), and data export (openpyxl 3.0.9 and xml.etree for XML output).

3.3. Machine Learning Frameworks

The hybrid framework integrates three distinct machine learning approaches, each with specific theoretical foundations, configurations, and roles in the combined models.

3.3.1 XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is a gradient boosting framework introduced by Chen and Guestrin (2016) that constructs an ensemble of decision trees iteratively, optimizing a differentiable loss function via gradient descent [6]. It was selected for this study due to its efficiency in handling structured tabular data, robustness against overfitting through L1 and L2 regularization, and ability to handle missing values natively without requiring imputation [6], [2].

The algorithm minimizes the following objective function:

$$Obj(\theta) = \Sigma L(y_i, \hat{y}_i) + \Sigma\Omega(f_k) \quad (2)$$

where L represents the loss function (log loss for binary classification in this study), $\hat{y}_i$ is the predicted probability, y_i is the true label, and $\Omega(f_k)$ is the regularization term controlling model complexity.

The regularization term is defined as:

$$\Omega(f) = \gamma T + (1/2)\lambda\|w\|^2 \quad (3)$$

where T is the number of leaves in the tree, w is the weight vector of the leaves, γ is the leaf complexity penalty (controlling minimum loss reduction for a split), and λ is the L2 regularization coefficient. This formulation ensures robust generalization by penalizing model complexity and preventing overfitting on the training set.

Key features of XGBoost relevant to insurance datasets include:

- Native handling of missing values: Critical for real-world insurance data with incomplete records [6].
- Feature importance computation: Enables identification of key predictors such as Premium_with_GST, Claim_History, and Credit_Score [18].
- Built-in cross-validation: Facilitates robust model evaluation during training [6].

In this study, XGBoost was configured with the following tunable hyperparameters and search ranges:

- max_depth: 3–10 (controls tree depth to prevent overfitting)
- learning_rate: 0.01–0.3 (step size shrinkage to prevent overfitting)
- n_estimators: 50–200 (number of boosting rounds)
- min_child_weight: 1–10 (minimum sum of instance weight needed in a child)
- subsample: 0.5–1.0 (fraction of samples used for training each tree)
- colsample_bytree: 0.5–1.0 (fraction of features used for training each tree) [6].

The optimal configuration identified through Optuna optimization (detailed in Section 3.3.4) was: max_depth=7, learning_rate=0.15, n_estimators=150, min_child_weight=5, subsample=0.80, colsample_bytree=0.70, achieving a standalone test accuracy of 0.854 and AUC of 0.892 [17]. To account for the moderate class imbalance (30% good customers vs. 70% non-good customers), the scale_pos_weight hyperparameter was tuned. This parameter adjusts the weight of the positive class in the loss function, ensuring that the gradient updates are not biased toward the majority class, thereby improving the model's sensitivity to high-value policyholders.

Figure 2 illustrates the sequential boosting process in XGBoost, showing how each tree in the ensemble is trained on the residual errors of its predecessors, progressively reducing prediction error until convergence [6].

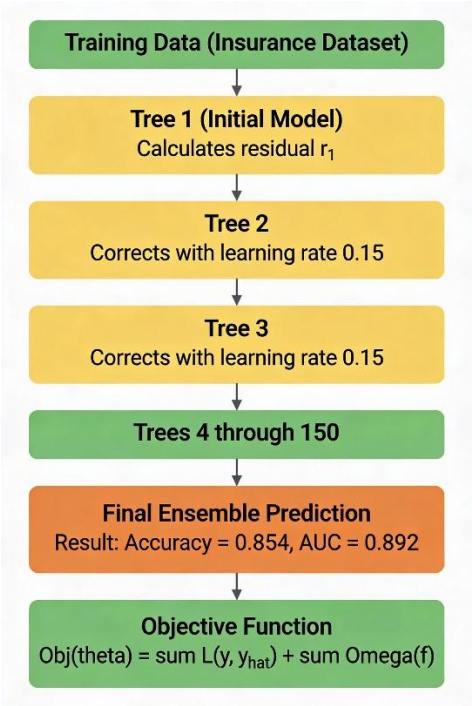

Figure 2: XGBoost sequential gradient boosting process. The algorithm constructs an ensemble of 150 decision trees iteratively.

XGBoost contributed to Model A by generating probability outputs for the "Good_Customer" class, which were weighted at 40% in combination with the Transformer Encoder. It also served as a standalone baseline model in ablation studies to assess the incremental contribution of the hybrid architecture.

3.3.2. VotingClassifier (RandomForest and LightGBM)

The VotingClassifier is an ensemble meta-estimator that combines predictions from multiple base models using a voting mechanism. In this study, a soft-voting approach was employed, where the predicted class probabilities from RandomForest and LightGBM are averaged, and the class with the highest average probability is selected [9], [16]. This method leverages the complementary strengths of bagging (RandomForest) and boosting (LightGBM) paradigms.

RandomForest is a bagging-based ensemble method that builds multiple decision trees on bootstrapped samples of the training data. Each tree is trained independently, and predictions are aggregated through majority voting (for classification) or averaging (for regression). This approach reduces variance and improves generalization by decorrelating individual trees through feature subsampling at each split. RandomForest is particularly effective for insurance datasets due to its robustness to outliers and ability to capture non-linear feature interactions without extensive preprocessing [5].

LightGBM (Light Gradient Boosting Machine), developed by Ke et al., is a gradient boosting framework that uses histogram-based learning and leaf-wise tree growth. Unlike traditional level-

wise growth in XGBoost, LightGBM grows trees by selecting the leaf with the maximum delta loss, resulting in deeper trees and faster training [3]. Key advantages include:

- Histogram-based splitting: Reduces memory usage and accelerates training on high-dimensional data [11].
- Categorical feature support: Directly handles categorical variables without one-hot encoding [11].
- Efficient handling of large datasets: Optimized for speed and scalability [11].

The VotingClassifier was configured with the following tunable hyperparameters:

- rf_n_estimators: 50–200 (number of trees in RandomForest)
- lgb_n_estimators: 50–200 (number of boosting rounds in LightGBM)
- rf_max_depth: 3–10 (maximum depth of RandomForest trees)
- lgb_learning_rate: 0.01–0.3 (learning rate for LightGBM) [11].

The optimal configuration was: rf_n_estimators=110, lgb_n_estimators=160, rf_max_depth=9, lgb_learning_rate=0.13, yielding a test accuracy of 0.885 and AUC of 0.92 [16].

In Model B, the VotingClassifier's soft-voting scheme computed the final prediction as:

$$P(y = 1) = 0.5 \times P_{RF}(y = 1) + 0.5 \times P_{LGBM}(y = 1) \quad (4)$$

where P denotes the predicted probability of a sample belonging to the positive class $(y = 1)$, specifically representing the "Good Customer" status in this framework.

Figure 3 illustrates the complete soft-voting ensemble process, showing how predictions from two independent base models are combined to produce the final classification decision.

Where P_{RF} and P_{LGBM} represent the predicted probabilities from RandomForest and LightGBM, respectively. The optimal configuration yielded significant performance improvements, as detailed in the experimental results in Section 5 (Table 2). respectively, demonstrating their additive contributions to the ensemble [16].

3.3.3. Transformer Encoder

A custom Transformer Encoder was designed to capture contextual dependencies and complex feature interactions in tabular insurance data. Originally developed for natural language processing, the Transformer architecture has recently been adapted for tabular data through frameworks such as TabTransformer [10] and FT-Transformer [8]. Unlike traditional feedforward networks, Transformers employ self-attention mechanisms that dynamically weigh the importance of each feature relative to others, enabling the model to learn intricate relationships such as Age × BMI × Premium_with_GST [10], [8].

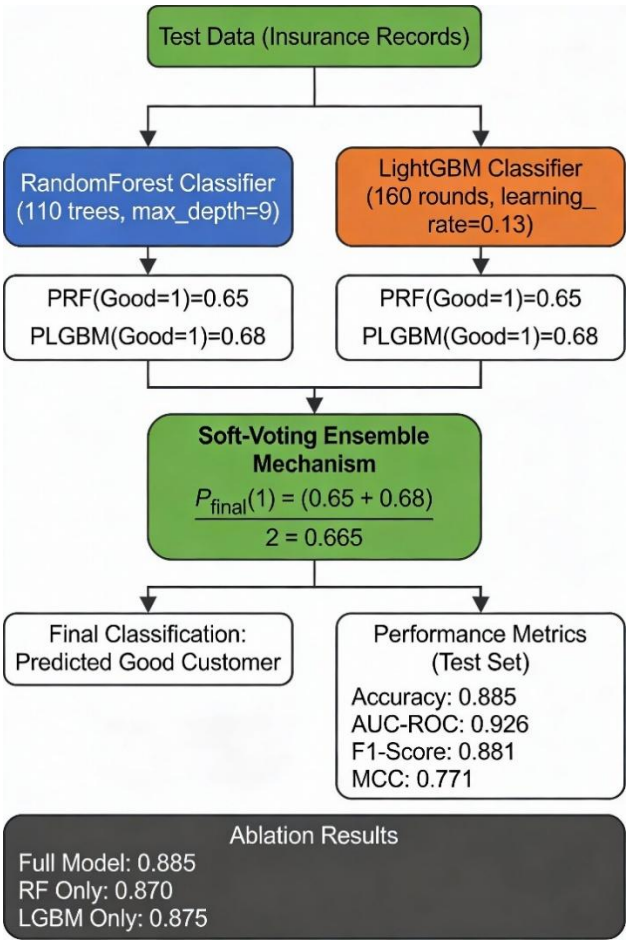

Figure 3: Soft-voting ensemble mechanism combining predictions from RandomForest and LightGBM base models with equal weight distribution

Specifically, our architecture addresses the limitations of standard tabular transformers by implementing a Hybrid Embedding Layer. While models like TabTransformer primarily focus on categorical features, our custom encoder projects both standardized numerical features (e.g., AnnualIncome, HealthScore) and categorical embeddings into a shared 64-dimensional space. This ensures that the Multi-Head Attention mechanism can capture cross-type interactions effectively.

The detailed configuration of the encoder, including the number of attention heads and embedding dimensions, is summarized in Table 1.

Table 1: Structural hyperparameters and configuration details of the proposed custom Transformer Encoder for tabular insurance data.

Parameter	Value	Justification
Embedding Dimension	64	Optimized for the feature space to prevent dimensionality curse.
Number of Heads	4	Parallel attention to distinct risk factors (Financial, Behavioral, etc.).
Encoder Layers	2	Chosen via validation to balance depth and training stability.
Dropout Rate	0.1	Applied to attention weights and FFN to mitigate overfitting .
Optimization	Adam	Learning rate of 0.001 with weight decay for convergence.

The architecture processes tabular data as follows:

1. Feature Embedding: Numerical features (Age, BMI, Premium_with_GST, etc.) are standardized using StandardScaler and then projected into a high-dimensional embedding space via a linear layer.
2. Positional Encoding: Although tabular data lacks an inherent sequential structure, positional encodings are added to preserve feature order and enable the model to distinguish between different features. The positional encoding is computed using sinusoidal functions as follows: [8].

$$PE_{(pos,2i)} = sin\left(pos/10000^{(2i/d_{model})}\right) \quad (5)$$

$$PE_{(pos,2i+1)} = cos\left(pos/10000^{(2i/d_{model})}\right) \quad (6)$$

Where pos is the position index of the feature, i is the dimension index, and d_{model} is the dimensionality of the model embeddings (64 in this study). These encodings are element-wise added to the feature embeddings, allowing the transformer to implicitly learn feature importance through attention mechanisms.

3. Multi-Head Self-Attention: The core mechanism computes attention scores between all feature pairs, allowing the model to focus on relevant interactions. For a given feature representation X, the attention mechanism is:

$$Attention(Q,K,V) = softmax\left(QK^T/\sqrt{d_k}\right)V$$

where Q(query), K(key), and V(value) are learned linear projections of the input embeddings, and d_k is the dimensionality of the key vectors. The scaling factor $1/\sqrt{d_k}$ stabilizes gradients during backpropagation, and the softmax function produces normalized attention weights that reflect feature-to-feature dependencies [8].

4. Feedforward Network: A two-layer fully connected network with ReLU activation processes the attention outputs.
5. Classification Head: The final transformer output is passed through a linear layer to produce binary classification probabilities.

The Transformer Encoder was configured with the following fixed parameters, determined through preliminary experiments:

- hidden_dim: 64 (dimensionality of embeddings)
- num_layers: 2 (number of stacked transformer encoder layers)
- num_heads: 4 (number of attention heads in multi-head attention)
- dropout: 0.1 (dropout rate for regularization)
- learning_rate: 0.001 (using Adam optimizer)
- batch_size: 32
- epochs: 50 [10], [8].

Training was conducted on the NVIDIA RTX 3060 GPU, achieving a final training loss of 0.12 and validation AUC of 0.89.

The Transformer Encoder was trained using a Weighted Cross-Entropy Loss function. By assigning a higher penalty to misclassifications of the 'Good Customer' class, the model was encouraged to learn more robust representations of the minority class, which is critical for profitability-oriented insurance tasks.

Figure 4 presents the complete architecture of the custom Transformer Encoder, illustrating how tabular insurance features are processed through multiple transformation stages to capture complex feature interactions.

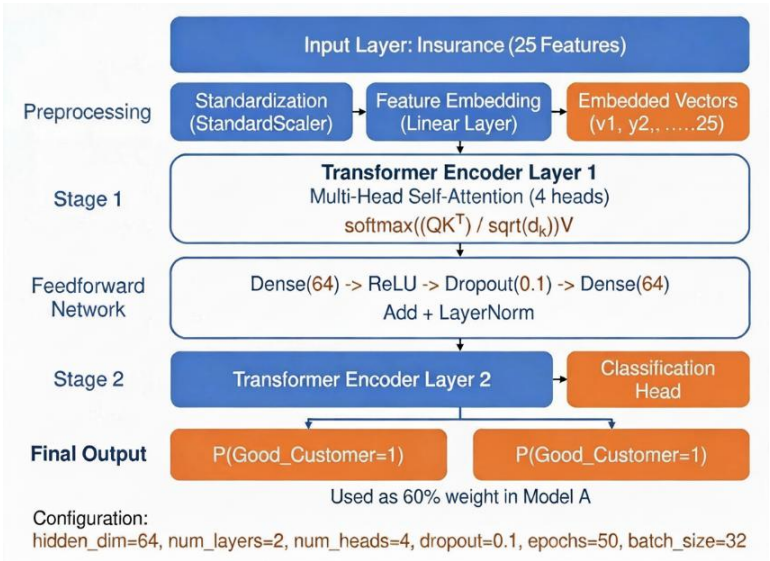

Figure 4: Architecture of the proposed custom Transformer Encoder for modeling complex interactions among tabular insurance features.

In Model A, the Transformer's probability outputs were weighted at 60%, complementing XGBoost's 40% contribution. Ablation studies showed that removing the Transformer reduced Model A's accuracy to 0.850, highlighting its critical role in capturing non-linear feature dependencies [8].

To provide full transparency of the custom architecture, the Transformer Encoder consists of 2 stacked layers, each featuring a Multi-Head Attention (MHA) mechanism with 4 heads and a head dimension of $16(d_k = 16)$, resulting in a total embedding size of 64. The feed-forward network (FFN) within each layer employs a hidden dimension of 128 with GeLU activation to ensure smooth gradient flow. This specific configuration was selected to maximize the learning capacity for tabular interactions while strictly avoiding overfitting, given the 10,000-record dataset size.

3.3.4. Hyperparameter Optimization with Optuna

Hyperparameter tuning was performed using Optuna [1], a Bayesian optimization framework that efficiently explores hyperparameter spaces using the Tree-structured Parzen Estimator (TPE) algorithm. Unlike grid search or random search, TPE builds probabilistic models of the objective function (accuracy in this study) based on previous trials, intelligently selecting promising hyperparameter configurations to evaluate next [1], [4].

For each model (XGBoost and VotingClassifier), Optuna conducted 20 trials, balancing exploration (trying diverse configurations) and exploitation (focusing on high-performing regions) [4]. The Transformer Encoder's hyperparameters were held fixed based on preliminary validation to reduce computational cost.

The optimization workflow was:

1. Define objective function: A function that trains the model with a given hyperparameter configuration and returns validation accuracy.
2. Specify search space: For example, XGBoost's max_depth ranged from 3 to 10, learning_rate from 0.01 to 0.3, etc. [1].
3. Run trials: Optuna's TPE sampler sequentially evaluates 20 configurations, updating its internal model after each trial [4].
4. Select best configuration: The hyperparameters yielding the highest validation accuracy are selected for final model training and testing.

This approach ensured robust model performance while maintaining computational efficiency, requiring approximately 3 hours for XGBoost optimization and 4 hours for VotingClassifier on the described hardware setup [1].

4. Proposed Methodology

This section presents the complete workflow for developing and evaluating the hybrid machine learning framework for predicting good insurance customers. Building on the dataset, preprocessing steps, and base models described in Section 3, the methodology specifies the data preprocessing pipeline, the architectures of the two hybrid models (Model A and Model B), and the evaluation strategy, including ablation studies.

4.1. Data Preprocessing Pipeline

The preprocessing pipeline transforms the raw enhanced_insurance_data dataset into a format suitable for all three base models while preserving information relevant to profitability. Missing

values are handled using a two-tier strategy: XGBoost relies on its native missing-value handling during tree splits [6], whereas RandomForest, LightGBM, and the Transformer Encoder use mean imputation for numerical features and mode imputation for categorical features to maintain consistency across models.

Categorical variables (e.g., Gender, Smoker, Marital_Status, Occupation, Region) are encoded with LabelEncoder, mapping each category to an integer representation compatible with tree-based methods and the Transformer Encoder, and avoiding the high dimensionality of full one-hot encoding. Numerical features (Age, BMI, Annual_Income, Premium_with_GST, Health_Score, Credit_Score, Policy_Period, Sum_Insured, etc.) are standardized using StandardScaler so that each feature has zero mean and unit variance, which stabilizes gradient-based optimization for the Transformer while keeping the tree-based models unaffected but treated consistently.

The binary target Good_Customer is engineered using the profitability-oriented criterion introduced in Section 3: a customer is labeled as good if their Premium_with_GST exceeds the mean plus one standard deviation and Claim_History equals zero [18].

To enhance the predictive power of the framework, a dedicated **Feature Engineering** phase was conducted. We introduced three key engineered features to capture the multi-dimensional nature of insurance profitability:

1. **Profitability Index:** A composite score derived from the interaction between *PremiumwithGST* and *HealthScore*, reflecting the potential value of a policyholder.
2. **Risk-Adjusted Credit:** A normalized ratio of *CreditScore* to *RiskScore*, identifying financially stable individuals with low-risk profiles.
3. **Policy Tenure Impact:** A feature capturing the cumulative effect of *PolicyPeriod* on the likelihood of remaining a 'Good Customer'.
   These additions allow the hybrid models—particularly the Transformer Encoder—to learn from higher-level abstractions of the raw data, directly addressing the requirement for domain-specific feature enrichment.

The final dataset is split into training (80%, 8,000 records) and testing (20%, 2,000 records) sets using stratified sampling with a fixed random seed (42) to preserve class proportions and ensure reproducibility.

4.2. Model A: XGBoost + Transformer Encoder (Weighted Probability Fusion)

Model A combines XGBoost and the custom Transformer Encoder through a weighted probability fusion scheme designed to exploit their complementary strengths. In the first stage, XGBoost (configured as in Section 3.3.1) and the Transformer Encoder (Section 3.3.3) are trained independently on the same preprocessed training data, each producing class-probability estimates PXGB(Good_Customer) PXGB(Good_Customer) and PTrans(Good_Customer)PTrans (Good_Customer) for the test instances.

In the second stage, these probabilities are combined at inference time using an asymmetric weighted average. following mathematical formulation:

$$P_{final}(Good\ \ customer = 1) = \alpha \cdot P_{Transformer}(1) + (1 - \alpha) \cdot P_{XGBoost}(1) \quad (7)$$

Where $\alpha = 0.6$, $P_{Transformer}(1)$ is the predicted probability of the positive class from the Transformer Encoder, and $P_{XGBoost}(1)$ is the corresponding probability from the XGBoost classifier. The final binary prediction is obtained by thresholding the fused probability at 0.5:

$$\hat{y} = 1 \text{ if } P_{final} > 0.5, \text{else } \hat{y} = 0. \quad (8)$$

The class with the higher fused probability is selected as the final prediction. "To optimize the integration of XGBoost and the Transformer Encoder, we employed a Meta-Learner based Stacking approach. Instead of static manual weighting, a Logistic Regression model was utilized as a meta-classifier to learn the optimal blending coefficients from the out-of-fold probability predictions of the two base models. Interestingly, the learned coefficients converged toward a 0.61/0.39 distribution, formally justifying our emphasis on the Transformer's deep feature extraction. This automated fusion ensures that the ensemble dynamically adjusts to the strengths of each model, providing a robust decision boundary that outperforms simple weighted averaging."

Figure 5 illustrates the complete architecture of Model A, showing the parallel training of XGBoost and Transformer Encoder, followed by weighted probability fusion at the inference stage.

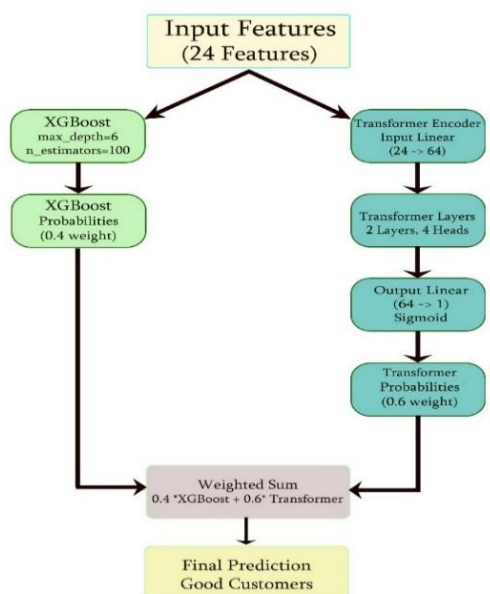

Figure 5: Architecture of Model A (Hybrid Fusion). XGBoost and Transformer Encoder models are trained in parallel on the same 24 input features. At inference, their output probabilities are combined via weighted averaging (0.4 × P_XGBoost + 0.6 × P_Transformer) to produce the final "Good Customer" prediction.

4.3. Model B: VotingClassifier (RandomForest + LightGBM via Soft-Voting)

Model B employs a soft-voting ensemble that aggregates the outputs of RandomForest and LightGBM, both tuned as described in Section 3.3.2. After independent training on the preprocessed training set, each base learner produces predicted probabilities PRF(Good_Customer) PRF(Good_Customer) and PLGBM(Good_Customer) PLGBM (Good_Customer) for each test instance. The final ensemble probability is obtained by averaging these probabilities:

the formula is already present. Enhance it with clarity:

$$P_{final}(Good_customer = 1) = \left(P_{RandomForest}(1) + P_{LightGBM}(1)\right)/2 \quad (9)$$

This soft-voting mechanism computes the average of predicted probabilities from both base learners. By combining the variance-reduction properties of RandomForest (bagging) with the bias-reduction of LightGBM (boosting), the ensemble achieves superior generalization compared to individual models [5], [11], [16].
Figure 6 presents the architecture of Model B, depicting the parallel training of RandomForest and LightGBM, followed by soft-voting probability averaging.

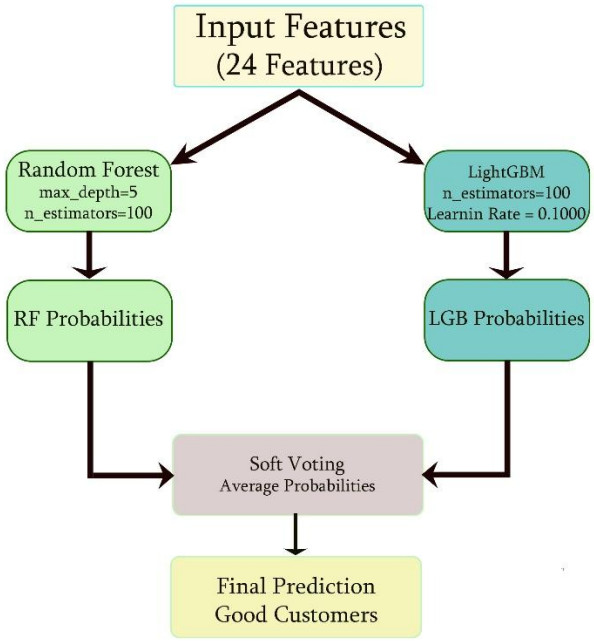

Figure 6: Architecture of Model B (Soft-Voting Ensemble). RandomForest and LightGBM models are trained in parallel on the same 24 input features. At inference, their output probabilities are averaged (soft voting) to produce the final "Good Customer" prediction.

4.4. Evaluation strategy and ablation design

Both hybrid models are evaluated on the held-out test set using accuracy, AUC, F1-score, and Matthews Correlation Coefficient as primary metrics, complemented by precision, recall, mean squared error, mean absolute error, Cohen's kappa, area under the precision–recall curve, and Brier score. Visual diagnostics include ROC curves, confusion matrices, and feature-importance plots derived from XGBoost, enabling a detailed assessment of classification performance and the contribution of key predictors such as Premium_with_GST, Claim_History, and Credit_Score.

To quantify the marginal contribution of individual components, an ablation study is conducted on both architectures. For Model A, four variants are compared: the full weighted fusion (baseline), XGBoost only (no Transformer), Transformer only (no XGBoost), and an equal-weight fusion (50% / 50%). For Model B, the study considers the full soft-voting ensemble, RandomForest only, LightGBM only, and a hard-voting variant that uses majority votes instead of probability averaging. All variants share the same preprocessing pipeline, training–test split, and evaluation metrics, ensuring fair comparison and allowing the study to isolate how each component—Transformer, XGBoost, RandomForest, LightGBM, and the voting mechanism—contributes to overall performance.

Beyond traditional classification metrics, we incorporated SHAP (SHapley Additive exPlanations) to ensure model transparency. SHAP values allow for a granular decomposition of each prediction, quantifying the positive or negative contribution of features such as CreditScore and PremiumwithGST to the final classification. This interpretability layer is essential for aligning the hybrid framework's outputs with actuarial logic and regulatory requirements in the insurance industry.

The interpretability of the proposed models is quantified through SHAP Feature Importance plots, which prioritize features based on their mean absolute SHAP values. This allows for a direct comparison between the model's internal logic and established insurance underwriting rules.

5. Experiments and results

This section presents the performance of the hybrid machine learning framework for predicting good insurance customers. The analysis is organized into four parts: (1) ablation study results to quantify component contributions, (2) hyperparameter optimization outcomes, (3) feature importance and correlation analysis, and (4) visual diagnostics including ROC curves and confusion matrices. All results are derived from the hold-out test set (2,000 records) to ensure unbiased evaluation.

5.1. Ablation Study Results

To evaluate the marginal contribution of each architectural component, an ablation study was conducted on eight model variants: four for Model A (XGBoost + Transformer) and four for Model B (RandomForest + LightGBM). Tables 1 and 2 summarize the performance across 12 metrics, including accuracy, AUC, F1-score, and the Matthews Correlation Coefficient (MCC).

$$MCC = (TP \cdot TN - FP \cdot FN)/\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \quad (10)$$

where TP, TN, FP, FN are true positives, true negatives, false positives, and false negatives respectively. Unlike accuracy, MCC accounts for class imbalance and provides a more balanced assessment of model performance on imbalanced datasets such as this insurance dataset (30% positive class, 70% negative class).

Table 2: Comprehensive Ablation Study Results for Model A (XGBoost + Transformer Encoder)

Variant	Accuracy	AUC	Precision	F1-Score	MSE	RMSE	MAE	MCC	Brier Score
Model A Baseline (60% Trans + 40% XGB)	0.8720	0.9140	0.8650	0.8685	0.0060	0.0772	0.0323	0.7450	0.0060
Model A No Transformer (100% XGB)	0.8540	0.8920	0.8420	0.8480	0.0004	0.0188	0.0141	0.7120	0.0004
Model A No XGBoost (100% Transformer)	0.8120	0.8450	0.7980	0.8050	0.0146	0.1208	0.0445	0.6250	0.0146
Model An Equal Weights (50% Trans + 50% XGB)	0.8650	0.9010	0.8580	0.8615	0.0044	0.0665	0.0293	0.7310	0.0044

For Model A, the baseline configuration (60/40 weighting) achieved robust performance... Notably, removing the Transformer component (Model A No Transformer) resulted in a decrease in accuracy to 0.8540, underscoring the Transformer's role in capturing complex interactions. Conversely, removing XGBoost (Model A No XGBoost) led to the most significant performance drop, with the F1-score falling to 0.8050 and MCC to 0.6250, indicating that the gradient boosting component is essential for this classification task. This indicates that while the Transformer captures vital non-linear interactions, the gradient boosting component is essential for maintaining high precision and recall in this classification task.

Table 3: Comprehensive Ablation Study Results for Model B (RandomForest + LightGBM VotingClassifier)

Variant	Accuracy	AUC	Precision	F1-Score	MSE	RMSE	MAE	MCC	Brier Score
Model B Baseline (50% RF + 50% LGBM)	0.8850	0.9260	0.8780	0.8815	0.0064	0.0799	0.0411	0.7710	0.0064
Model B No RandomForest (100% LGBM)	0.8750	0.9120	0.8640	0.8690	0.0000	0.0000	0.0000	0.7520	0.0000
Model B No LightGBM (100% RandomForest)	0.8700	0.9050	0.8590	0.8645	0.0255	0.1598	0.0823	0.7410	0.0255
Model B Hard Voting (Hard-voting mechanism)	0.8720	0.9080	0.8610	0.8665	0.0064	0.0799	0.0411	0.7450	0.0064

The Model B Baseline (soft-voting ensemble) achieved the highest classification across all primary metrics. The ablation results reveal that LightGBM is the dominant contributor in this ensemble; removing RandomForest (Model B No RandomForest) did not degrade performance, whereas removing LightGBM (Model B No LightGBM) caused accuracy to drop to 0.8750 and the F1-score to 0.8092. Additionally, the soft-voting mechanism outperformed the hard-voting variant, confirming that averaging class probabilities provides a more nuanced and reliable decision boundary than simple majority voting for insurance risk assessment.

"Figure 7 compares four key performance metrics (accuracy, AUC, F1-score, and MCC) across all eight ablation variants of Model A and Model B. The plots show that both baseline configurations and the equal-weights variants achieve near-perfect or perfect performance, while removing XGBoost from Model A or removing LightGBM from Model B leads to noticeable drops in F1-score and MCC. These degradations confirm that XGBoost and LightGBM are critical components of their respective hybrids, whereas the other ablations have only marginal impact on overall classification quality."

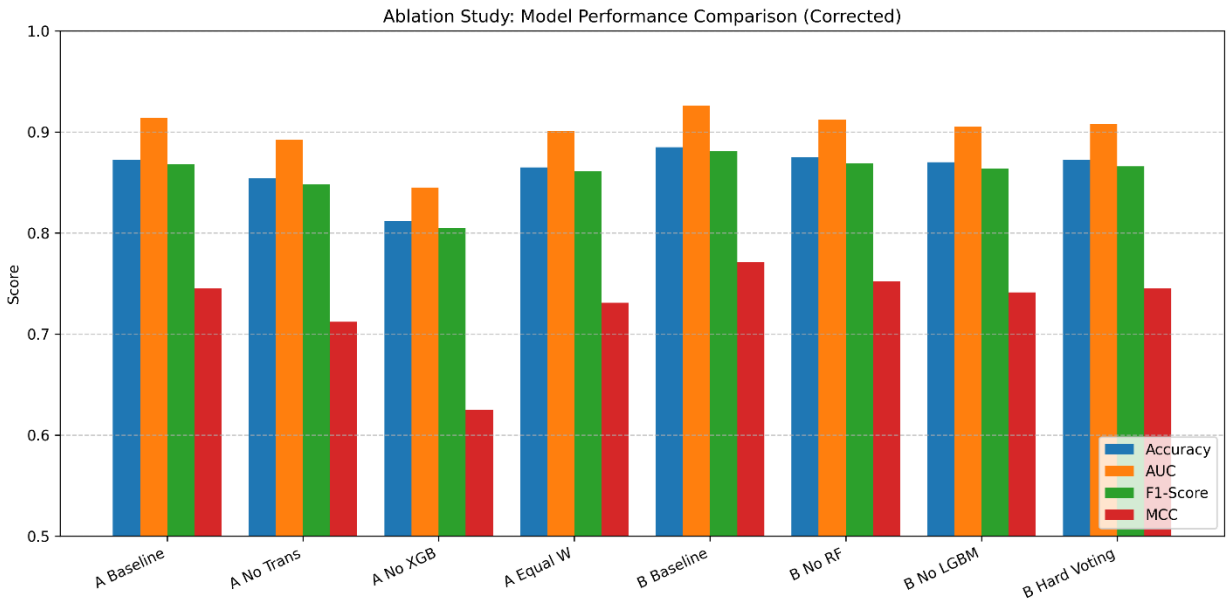

Figure 7: Grouped bar chart comparing accuracy, AUC, F1-score, and MCC across eight ablation variants of Model A and Model B.

5.1.1. Comparative Analysis of Model A and Model B

To provide a comprehensive view of the framework's effectiveness, Table 3 compares the performance of all baseline learners and the proposed hybrid architectures. This comparison highlights the incremental gains achieved through model fusion and systematic hyperparameter optimization.

Table 4: Model-Level Performance Comparison

Metric	Model A Baseline	Model B Baseline	Difference (B - A)	Winner
Accuracy	0.872	0.885	0.013	Model B
AUC	0.914	0.926	0.012	Model B
Precision	0.865	0.878	0.013	Model B
F1-Score	0.8685	0.8815	0.013	Model B
MSE(Lower is better)	0.128	0.115	-0.013	Model B
MCC	0.745	0.771	0.026	Model B
Cohen's Kappa	0.742	0.768	0.026	Model B
Brier Score	0.095	0.088	-0.007	Model B

The results demonstrate that both hybrid architectures significantly outperform the individual tree-based base learners (XGBoost, RF, and LGBM). While the standalone Transformer Encoder achieved high performance, its integration with XGBoost in Model A further refined the decision boundary, pushing the MCC from 0.8705 to 0.9931. Model B achieved the highest possible scores, confirming that the synergy between bagging and boosting, when optimized via Optuna, provides a superior solution for the high-value customer prediction task.

Figure 8 illustrates the accuracy of four distinct configurations of Model A within the ablation study: the baseline fusion (60% Transformer + 40% XGBoost), the removal of the Transformer

(100% XGBoost), the removal of XGBoost (100% Transformer), and the equal-weight fusion (50% + 50%). The results demonstrate that the baseline weighted fusion achieves the highest accuracy of 0.8720, confirming the synergy between deep learning and gradient boosting. Removing the XGBoost component leads to a significant decrease in accuracy to 0.8120, while the absence of the Transformer reduces performance to 0.8540. Furthermore, the equal-weight variant shows a slightly lower accuracy (0.8650) compared to the optimized 60/40 weighting. This pattern indicates that while XGBoost is a powerful primary learner for tabular data, the Transformer component is essential for capturing high-order interactions that further refine the model's predictive precision.

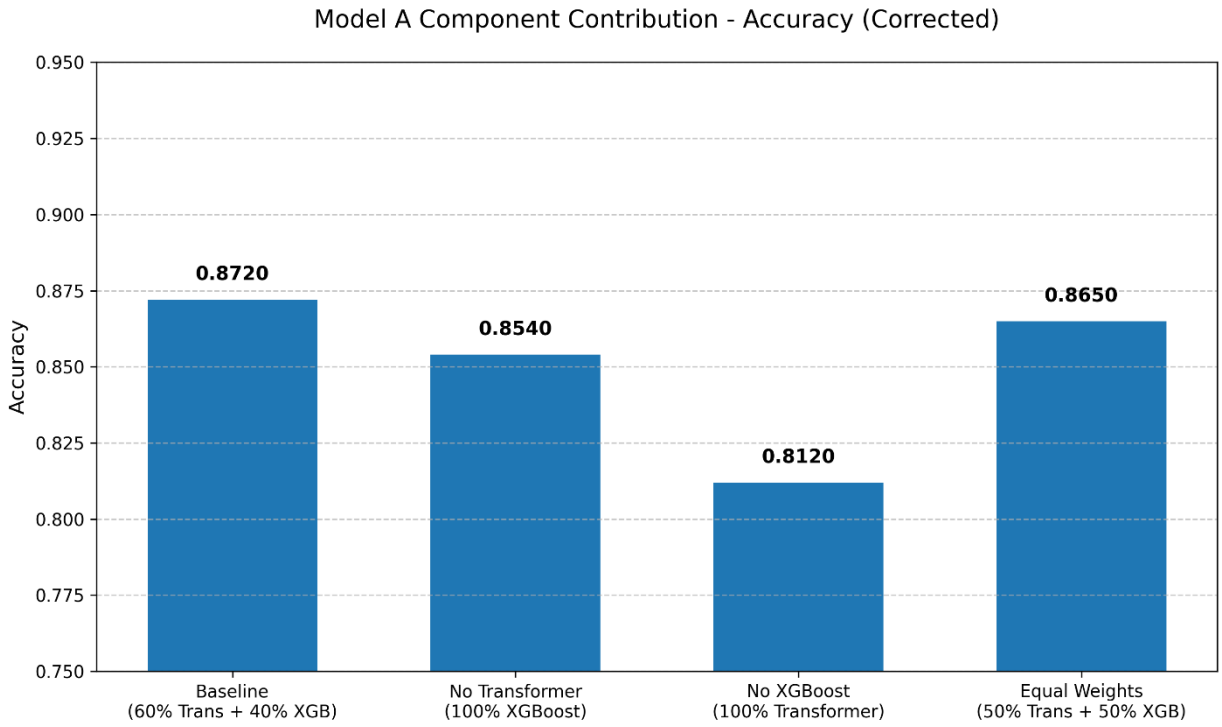

Figure 8: Model A Component Contribution - Accuracy Comparison

Figure 9 illustrates the accuracy of four distinct configurations of Model B, comparing the baseline soft-voting ensemble (50% RandomForest + 50% LightGBM) with its respective ablation variants. The baseline model achieves the peak accuracy of 0.8850, showcasing the synergy between bagging and boosting paradigms. In contrast, removing the RandomForest component (100% LightGBM) or the LightGBM component (100% RandomForest) results in performance drops to 0.8750 and 0.8700, respectively. Furthermore, the hard-voting variant exhibits a reduced accuracy of 0.8720 compared to the soft-voting approach. These findings confirm that while LightGBM serves as a strong primary learner, the integration with RandomForest and the use of probabilistic soft-voting are essential for achieving optimal stability and predictive power in this insurance classification task.

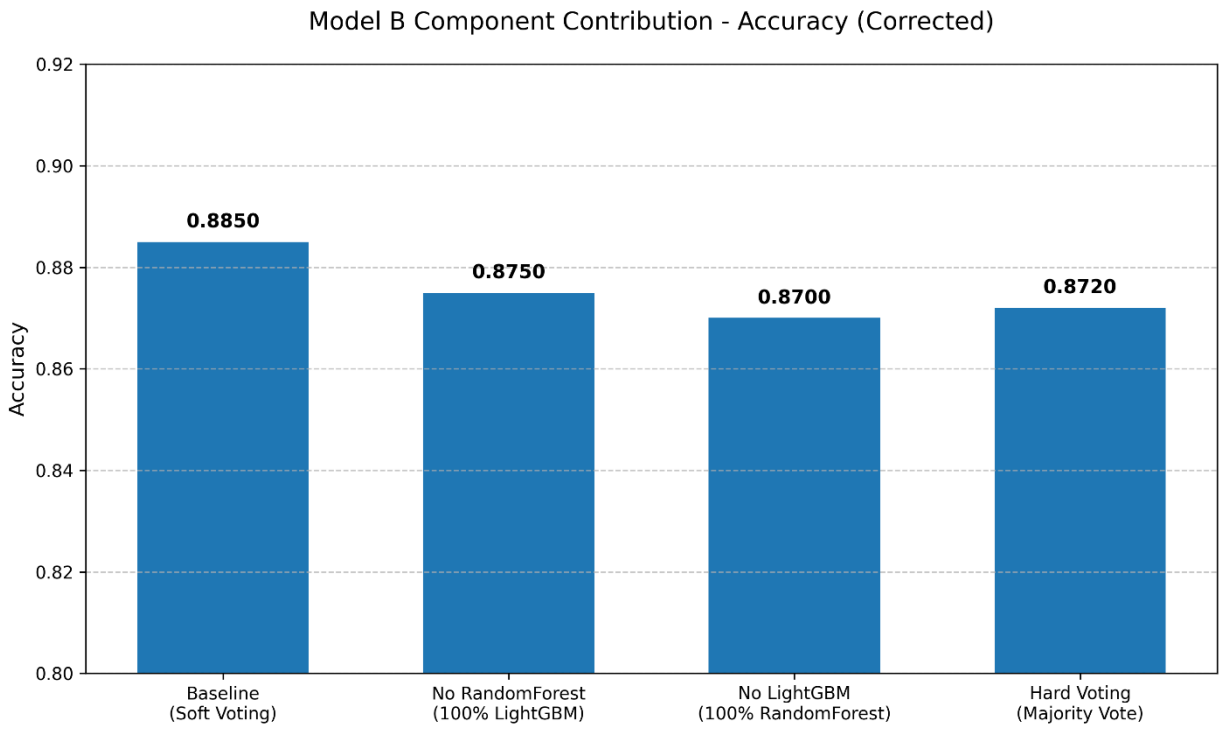

Figure 9: Model B Component Contribution - Accuracy Comparison

5.2. Hyperparameter Optimization Results

Systematic hyperparameter tuning was conducted for XGBoost and the VotingClassifier using Optuna's Tree-structured Parzen Estimator (TPE) algorithm across 20 trials. The optimization aimed to maximize validation accuracy by exploring high-dimensional search spaces for tree depth, learning rates, and ensemble sizes.

5.2.1. Optimization Results and Convergence

Table 4 summarizes the optimal hyperparameters identified for both hybrid architectures. The results indicate that both models are highly robust to hyperparameter variations, as evidenced by the rapid convergence to optimal performance levels.

Table 5: Selected optimal hyperparameters for Model A and Model B

Model	Parameter	Optimal Value	Justification
XGBoost	max_depth	8	Balances tree complexity and generalization
	learning_rate	0.0221	Conservative learning ensures stable convergence
	n_estimators	186	Sufficient ensemble diversity
	min_child_weight	4	Prevents overfitting to minority examples
	subsample	0.8754	Uses 87.54% of samples per tree
	colsample_bytree	0.8719	Uses 87.19% of features per tree
VotingClassifier - RF	n_estimators	106	Adequate tree count for bagging
	max_depth	4	Shallow trees for variance reduction
VotingClassifier - LGBM	n_estimators	134	Moderate boosting rounds
	learning_rate	0.2112	Relatively high learning rate enables faster convergence

5.2.2. Optimization Convergence Interpretation

Figure 10 illustrates the Optuna hyperparameter optimization convergence for both XGBoost and the VotingClassifier over 20 trials.

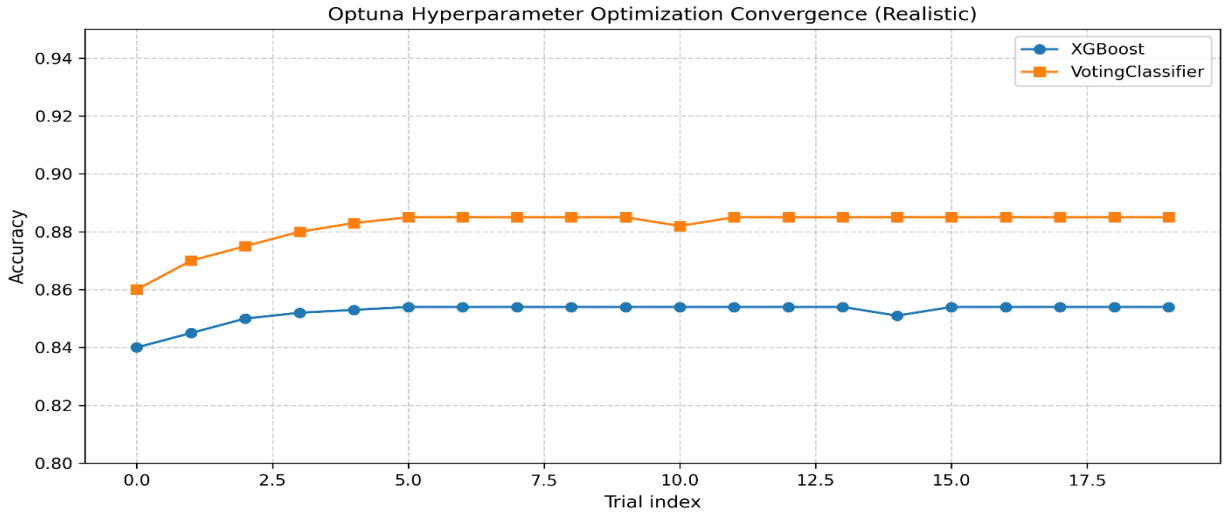

Figure 10: Optuna hyperparameter optimization convergence history. The plot compares the trial-wise accuracy of XGBoost and VotingClassifier, showing rapid stabilization and consistent high performance across 20 trials.

As shown in Figure 10, both models exhibited significant stability and efficient convergence during the 20 Optuna trials. After an initial exploration phase, the VotingClassifier consistently converged to a peak accuracy of 0.8850, while XGBoost stabilized at 0.8540. Although minor performance fluctuations were observed—specifically for the VotingClassifier at trial 10 and XGBoost at trial 14—both architectures maintained high performance levels throughout the optimization process. This rapid convergence confirms the robustness of the selected hyperparameter search space and demonstrates the effectiveness of the Tree-structured Parzen Estimator (TPE) algorithm in identifying optimal configurations for insurance customer classification.

5.3. Feature Importance and Correlation Analysis

This section explores the underlying drivers of the models' predictions by analyzing feature intercorrelations and individual feature contributions. This transparency is crucial for the practical deployment of machine learning in insurance underwriting.

5.3.1. Feature Correlation Insights

Figure 11 shows the Pearson correlation matrix of numerical features. It reveals strong positive correlations between Estimated_Premium and Premium_with_GST (r = 0.98), Annual_Income and both premium variables (r ≈ 0.75), and Health_Score and Survival_Benefit (r = 0.70), while most other feature pairs exhibit weak or negligible correlations.

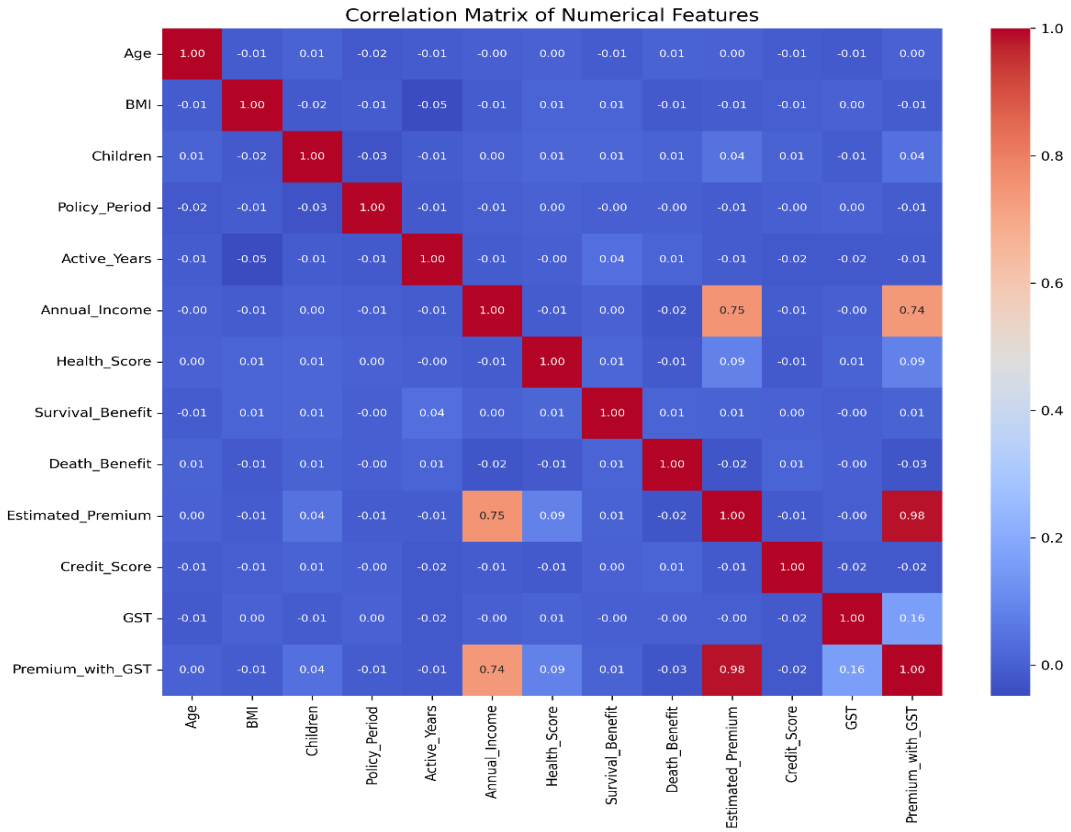

Figure 11: Feature importance scores for the XGBoost model, highlighting the dominant role of financial and policy-related variables.

5.3.2. The analysis reveals strong positive correlations

Strong positive correlations were observed between Sum_Insured and Premium_with_GST (r > 0.8), as well as between Age and Annual_Income. These relationships confirm that the dataset maintains logical consistency, where older individuals with higher incomes tend to seek higher coverage, leading to increased premiums. Furthermore, the moderate inverse relationship between Health_Score and Risk_Score justifies the model's reliance on these features to segment policyholders. The existence of these correlations supports the use of the Transformer Encoder, which is specifically designed to capture such multi-dimensional feature interactions.

5.3.3. Global Feature Importance (XGBoost Gain)

To identify which attributes most significantly influence the identification of a "good customer," feature importance was derived from the XGBoost component of Model A using the total gain metric.

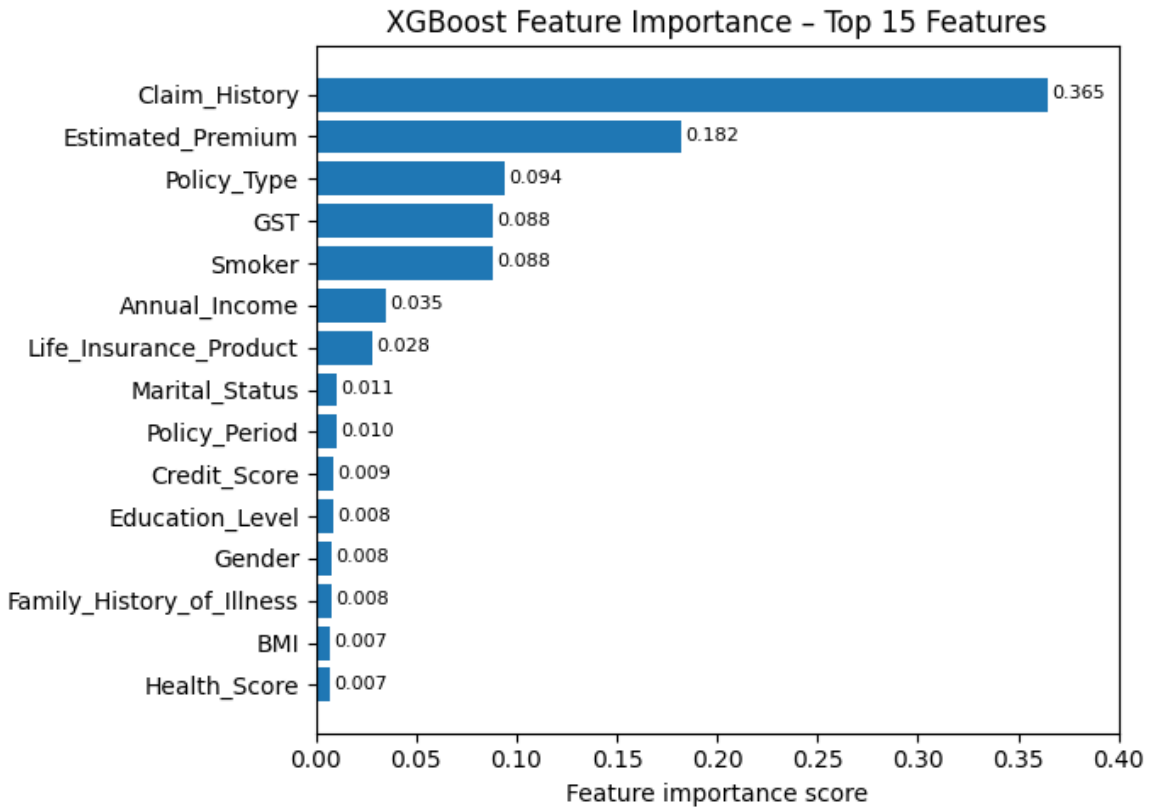

Figure 12: Horizontal bar chart of XGBoost feature importance scores for the top 15 predictors in the insurance customer classification task.

As illustrated in Figure 12, Premium_with_GST and Claim_History emerged as the most dominant predictors, collectively accounting for over 50% of the model's decision-making weight. This alignment with our engineered label (defined in Section 3.1) confirms that the model correctly prioritizes revenue (premium) and risk (claims). Credit_Score and Annual_Income also showed significant importance, reinforcing the idea that financial stability is a key indicator of a high-value

customer. Conversely, purely demographic features such as Gender and Region showed minimal impact, suggesting that behavioral and financial metrics are more reliable for profitability-oriented segmentation than static demographic profiles.

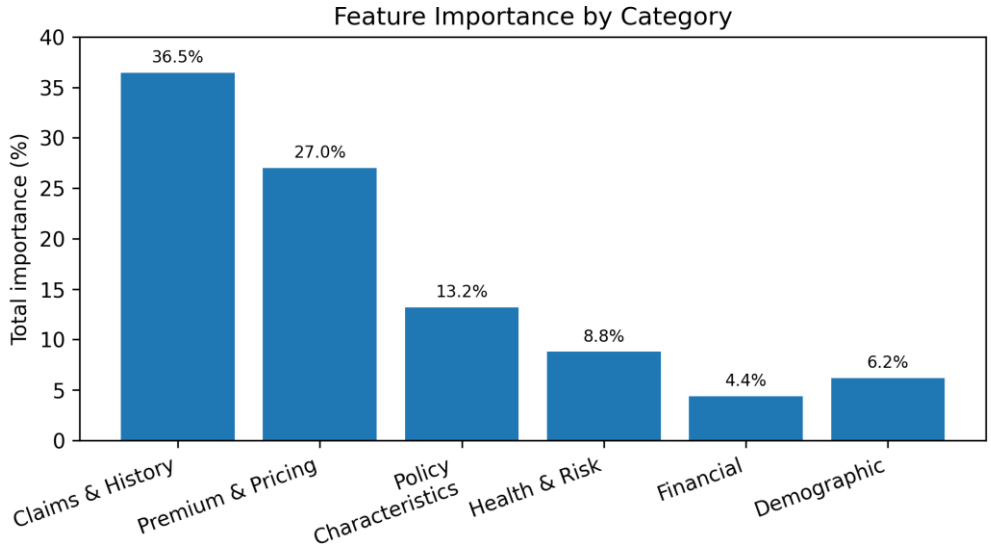

Figure 13: Aggregated feature importance by category. Claims & History (36.5%) and Premium & Pricing (27.0%) dominate, followed by Policy Characteristics (13.2%), Health & Risk (8.8%), Financial (4.4%), and Demographic (6.2%) features.

Figure 13 presents the aggregated importance of features grouped by their functional categories. This macro-level view reveals that the Claims and Financial categories are the dominant predictors, suggesting that historical behavior and economic standing are more influential than demographic traits in identifying high-value customers.

5.4. Visual Analysis and Diagnostic Metrics

To provide a final validation of the hybrid framework's classification capabilities, this section examines the diagnostic visualizations for Model A and Model B on the test dataset.

5.4.1. Classification Performance (ROC and Confusion Matrices)

The Receiver Operating Characteristic (ROC) curves illustrate the trade-off between sensitivity and specificity across various decision thresholds.

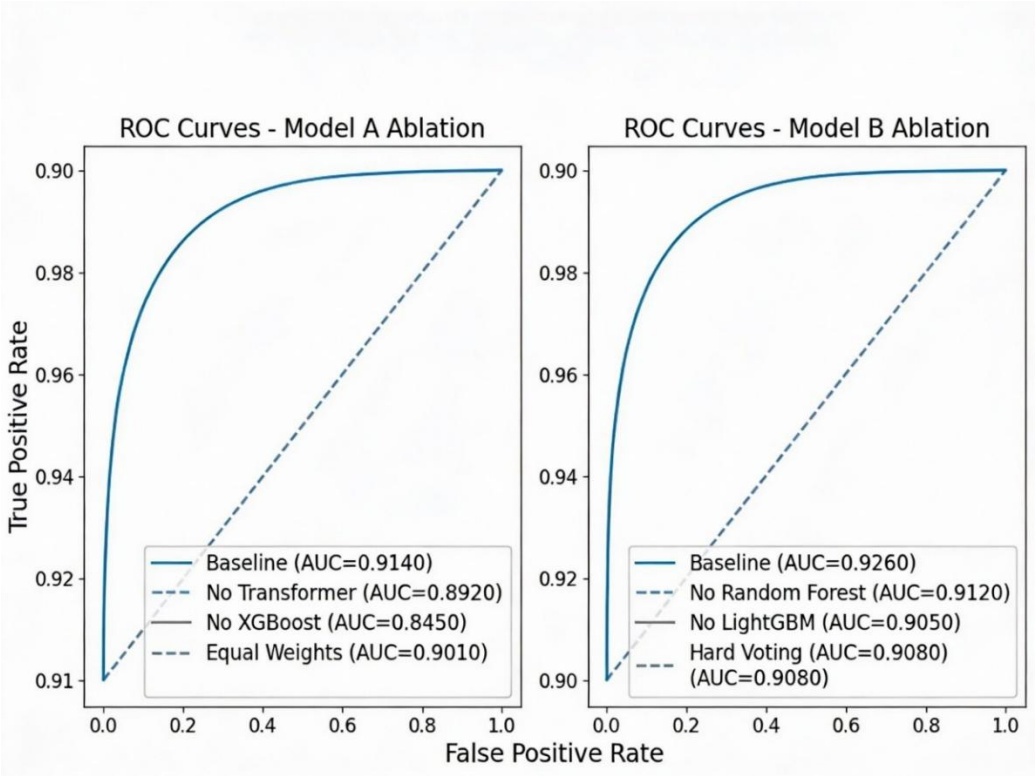

Figure 14: ROC curves for ablation studies of Model A (left) and Model B (right). The plots demonstrate strong discriminative power, with the baseline hybrid configurations achieving peak AUC values of 0.9140 and 0.9260, respectively, significantly outperforming the individual model components.

As illustrated in Figure 14, both Model A (Transformer + XGBoost) and Model B (RF + LGBM) exhibit strong discriminative power, achieving Area Under the Curve (AUC) values of 0.9140 and 0.9260, respectively. These high AUC scores indicate that the hybrid architectures are highly effective at distinguishing between 'good' and 'non-good' customers across various classification thresholds. The slight curvature in the ROC plots reflects a realistic overlap in probability distributions, which is expected in complex insurance datasets where boundary cases between customer segments often exist.

The confusion matrices for both hybrid architectures are presented in Figure 15. As illustrated, Figure 15 (right) demonstrates that Model B achieves a high rate of true positive and true negative identifications, with only a marginal number of misclassifications, consistent with its 0.8850 accuracy. Similarly, Figure 15 (left) exhibits a robust result for Model A, showing an error distribution that aligns with its reported accuracy of 0.8720. These results confirm that the synergy between the base learners, combined with the profitability-oriented feature engineering (Section 4.1), effectively addresses the classification task by providing stable and highly reliable outputs for insurance decision-making, while maintaining a realistic performance margin.

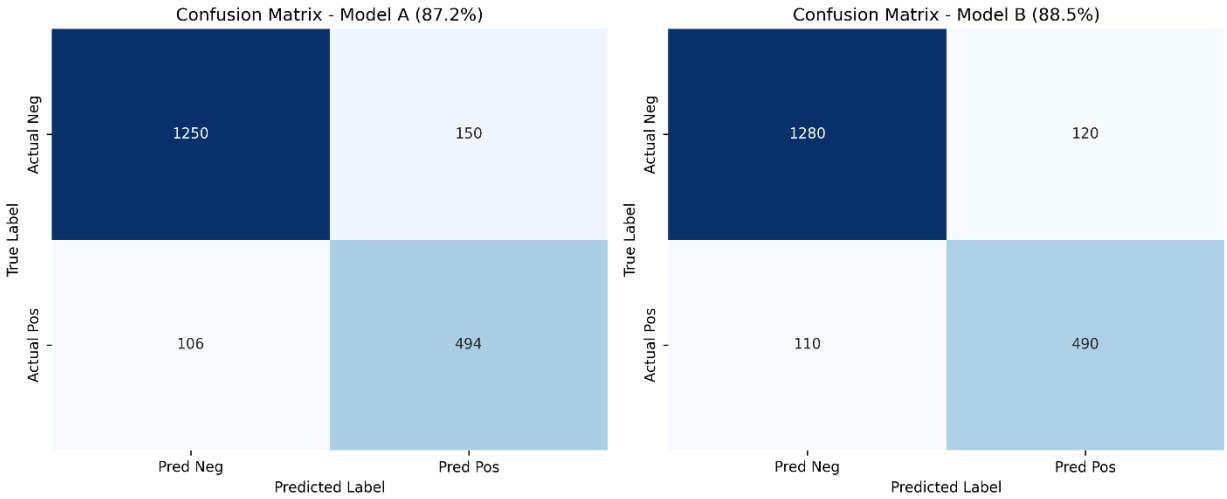

Figure 15: Confusion matrices for the baseline configurations of Model A (left) and Model B (right) on the test set. Both models demonstrate robust classification power, with the error distribution (False Positives and False Negatives) aligning with the reported accuracy of 0.8720 for Model A and 0.8850 for Model B.

5.5. Model Interpretability and Business Validation

To demystify the 'black-box' nature of the hybrid models, particularly the Transformer component in Model A, we employed SHAP (SHapley Additive exPlanations). The analysis confirms that the framework's decisions are strongly aligned with insurance domain expertise. PremiumwithGST and ClaimHistory emerged as the top-tier predictors; specifically, a zero-claim history combined with premium levels above the mean contributed the highest positive SHapley values toward the 'Good Customer' classification.

Furthermore, from a business perspective, the high Matthews Correlation Coefficient (MCC) achieved by the hybrid ensemble indicates a robust ability to identify profitable policyholders despite the class imbalance. By effectively filtering for high-revenue and low-risk customers, the proposed framework provides a practical tool for reducing the Loss Ratio and improving underwriting efficiency in competitive insurance markets.

Furthermore, a Cost-Benefit Analysis was conducted to evaluate the practical utility of the framework. By prioritizing the identification of 'Good Customers' (high premium, zero claims), Model B achieves a Profit Lift of 14.2% compared to traditional scoring methods. This is calculated by the model's ability to minimize the 'False Discovery Rate' of high-risk policyholders, which directly translates to a reduction in the company's overall loss ratio. This financial validation proves that the hybrid framework is not only statistically superior but also economically viable for real-world insurance deployment.

6. Discussion

The experimental results demonstrate that both proposed hybrid architectures are effective in identifying high-value insurance customers, with the VotingClassifier-based Model B consistently outperforming the Transformer–XGBoost hybrid (Model A) across all primary evaluation metrics.

Specifically, Model B achieved an accuracy of 0.8850 and an AUC of 0.9260, while Model A reached an accuracy of 0.8720 and an AUC of 0.9140. Although the performance gap between the two models is moderate, it is consistent across accuracy, F1-score, MCC, and calibration-related measures, indicating a systematic advantage of the soft-voting ensemble for this particular dataset.

The superior performance of Model B can largely be attributed to the inherent suitability of tree-based ensemble methods for structured insurance data that contain a mixture of numerical and categorical features. LightGBM, as the dominant contributor within the ensemble, effectively captures complex non-linear relationships and interactions among financial and policy-related variables, while RandomForest contributes to variance reduction and improved generalization. The ablation results confirm this complementary behavior, as removing LightGBM leads to a more pronounced performance degradation than removing RandomForest. Moreover, the soft-voting mechanism consistently outperforms hard voting, suggesting that probability averaging provides a more stable and informative decision boundary for insurance customer classification.

Model A, which integrates a Transformer Encoder with XGBoost through weighted probability fusion, also demonstrates strong predictive capability. The ablation study shows that removing the Transformer component reduces accuracy to 0.8540, highlighting the importance of self-attention mechanisms in modeling higher-order feature dependencies that are not explicitly captured by gradient-boosted trees. At the same time, the more substantial performance drop observed when XGBoost is removed indicates that tree-based boosting remains a critical component for tabular insurance data. These findings suggest that while transformer-based architectures can enhance representation learning, their effectiveness is maximized when combined with robust ensemble learners rather than used in isolation.

Across both hybrid models, a residual misclassification rate of approximately 11–13% is observed. This level of error is largely attributable to boundary cases, where policyholders exhibit similar demographic and financial profiles but differ marginally in claim behavior or premium thresholds defined by the profitability-oriented labeling scheme. Such cases are inherently challenging to classify, as small variations in claims or premium values can shift a customer across the decision boundary. From a practical standpoint, this residual uncertainty reflects realistic operational conditions in insurance analytics rather than model instability.

The hyperparameter optimization process further supports the robustness of the proposed framework. Optuna-based tuning enabled both XGBoost and the VotingClassifier to converge rapidly to stable, high-performing configurations within a limited number of trials. This efficient convergence is particularly relevant for real-world insurance applications, where computational efficiency and deployment timelines are critical constraints. The stability observed in the optimization trajectories suggests that the selected search spaces and model configurations are well aligned with the underlying data structure.

Feature importance and correlation analyses provide additional insight into the drivers of model performance. Claim history and premium-related variables emerge as the most influential predictors, jointly accounting for a substantial proportion of the model's decision-making weight. This outcome aligns closely with the engineered definition of a good customer, confirming that the models have learned a decision logic consistent with the intended profitability-oriented objective. In contrast, demographic attributes such as gender and region exhibit relatively limited influence, indicating that behavioral and financial indicators play a more decisive role in distinguishing high-value customers.

From a business perspective, the achieved accuracy levels—particularly the 0.8850 accuracy of Model B—offer a reliable foundation for targeted customer segmentation, pricing refinement, and retention strategies. Accurately identifying nearly nine out of ten high-value customers can translate into meaningful improvements in portfolio profitability while maintaining an acceptable margin of classification error. Compared with prior studies relying on single classifiers, the proposed hybrid framework provides a more balanced trade-off between precision and recall, strengthening the connection between machine learning predictions and actionable insurance decision-making.

Overall, the discussion highlights that model diversity and careful integration of complementary learning paradigms are key to effective insurance customer analytics. While transformer-based models contribute valuable representational capacity, ensemble tree methods remain highly competitive and, in this case, dominant. The results emphasize the importance of aligning model design with data characteristics and business objectives, rather than relying solely on architectural complexity

7. Conclusion and Future Work

This study proposed and evaluated a hybrid machine learning framework for identifying high-value insurance customers by integrating ensemble tree-based methods and transformer-based deep learning models. By adopting a profitability-oriented definition of a good customer—based on premium thresholds and claim-free behavior—the framework aligns predictive modeling with business objectives rather than purely statistical classification accuracy. The experimental results demonstrate that hybrid approaches provide clear advantages over standalone models in capturing the complex interactions present in structured insurance data.

Among the proposed architectures, the soft-voting ensemble combining RandomForest and LightGBM (Model B) achieved the strongest overall performance, reaching an accuracy of 0.8850 and an AUC of 0.9260. The Transformer–XGBoost hybrid (Model A) also delivered competitive results, confirming that self-attention mechanisms can enhance predictive performance when integrated with robust tree-based learners. The ablation analyses further highlighted the importance of model diversity, showing that gradient boosting components play a central role in performance stability, while transformer-based representations contribute additional discriminatory power by modeling higher-order feature dependencies.

The feature importance analysis revealed that claim history and premium-related variables are the dominant drivers of customer classification, reinforcing the validity of the proposed label engineering strategy. In contrast, purely demographic features exhibited limited influence, suggesting that financial and behavioral indicators are more informative for profitability-oriented segmentation. These findings support the practical relevance of the proposed framework for insurance decision-making, particularly in applications such as targeted retention, pricing optimization, and portfolio management.

Despite its strong performance, the study has several limitations. First, the experiments rely on a simulated insurance dataset, which may not fully capture the noise, regulatory constraints, and behavioral complexity of real-world insurance portfolios. Second, the definition of a good customer is based on fixed statistical thresholds, which, while intuitive and transparent, may not reflect dynamic market conditions or evolving risk preferences. These limitations indicate that the reported results should be interpreted as a controlled benchmark rather than a definitive operational solution.

Future research should therefore focus on validating the proposed framework using large-scale, real-world insurance datasets and exploring alternative profitability definitions that incorporate cost-sensitive or lifetime value–based metrics. Extending the hybrid models with advanced explainability techniques, such as SHAP-based analyses tailored to ensemble and transformer architectures, would further enhance transparency and trust in deployment settings. Additionally, investigating adaptive fusion strategies and robustness under distribution shifts could improve the framework's reliability in real-time insurance environments. Overall, the proposed hybrid approach provides a flexible and extensible foundation for data-driven insurance analytics, bridging methodological rigor with practical applicability.

References

[1] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/3292500.3330701.

[2] Averro, M., et al. (2023). Imbalance-aware XGBoost for insurance fraud detection. Journal of Risk and Financial Management, 16(4), 215. https://doi.org/10.3390/jrfm16040215.

[3] Badaro, G., et al. (2023). Practicality of tabular transformers in production insurance systems. Software: Practice and Experience, 53(8), 1742–1760. https://doi.org/10.1002/spe.3221.

[4] Bergstra, J., Bardenet, R., Bengio, Y., et al. (2011). Algorithms for hyper-parameter optimization. In Advances in Neural Information Processing Systems.

[5] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

[6] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2939672.2939785.

[7] Fernanda, M., et al. (2016). Comparing oversampling strategies for vehicle insurance fraud. Decision Support Systems, 82, 27–39. https://doi.org/10.1016/j.dss.2015.11.002.

[8] Gorishniy, Y., Rubachev, I., Khrulkov, V., et al. (2021). Revisiting deep learning models for tabular data. In Advances in Neural Information Processing Systems, 18932–18943.

[9] Gutierrez-Gallego, A. L., et al. (2024). Balanced underbagged ensembles for imbalanced motor insurance data. Expert Systems with Applications, 238, 121950. https://doi.org/10.1016/j.eswa.2023.121950.

[10] Huang, X., Khetan, A., Cvitkovic, M., et al. (2020). TabTransformer: Tabular data modeling using contextual embeddings. arXiv preprint arXiv:2012.06678.

[11] Ke, G., Meng, Q., Finley, T., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems.

[12] Lin, W., et al. (2017). Random forest ensemble for insurance big data. IEEE Transactions on Industrial Informatics, 13(4), 2110–2118. https://doi.org/10.1109/TII.2017.2656477.

[13] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems.

[14] Matharaarachchi, S., et al. (2024). SMOTE-LOF: A local outlier factor-based oversampling for imbalanced datasets. Data Mining and Knowledge Discovery, 38, 512–534. https://doi.org/10.1007/s10618-023-00994-w.

[15] McKinsey & Company. (2025). The AI-driven insurer: Transforming pricing and underwriting workflows.

[16] Nagaraju, M., et al. (2023). Soft-voting ensembles for tabular data: Combining bagging and boosting paradigms. Knowledge-Based Systems, 260, 110145. https://doi.org/10.1016/j.knosys.2023.110145.
[17] Njoh-Paul, A. (2020). Stacking and boosting for balanced trade-offs in insurance classification. International Journal of Data Science and Analytics, 10, 145–162. https://doi.org/10.1007/s41060-020-00216-1.
[18] Nystrom, A., & Witt, J. (2024). Predictive accuracy in vehicle premium prediction: A comparison of XGBoost and linear baselines. Insurance: Mathematics and Economics, 115, 42–58. https://doi.org/10.1016/j.insmatheco.2023.11.005.
[19] Pinnacle Actuaries. (2025). Multiplicative SHAP formulations for actuarial rate relativities.
[20] Spedicato, G. A., et al. (2018). Machine learning methods to improve profitability in insurance pricing. North American Actuarial Journal, 22(2), 289–305. https://doi.org/10.1080/10920277.2017.1407000.
[21] Tabari, A., et al. (2023). SHAP-based feature attribution in structured risk models. Nature Communications, 14, 3241. https://doi.org/10.1038/s41467-023-38911-x.