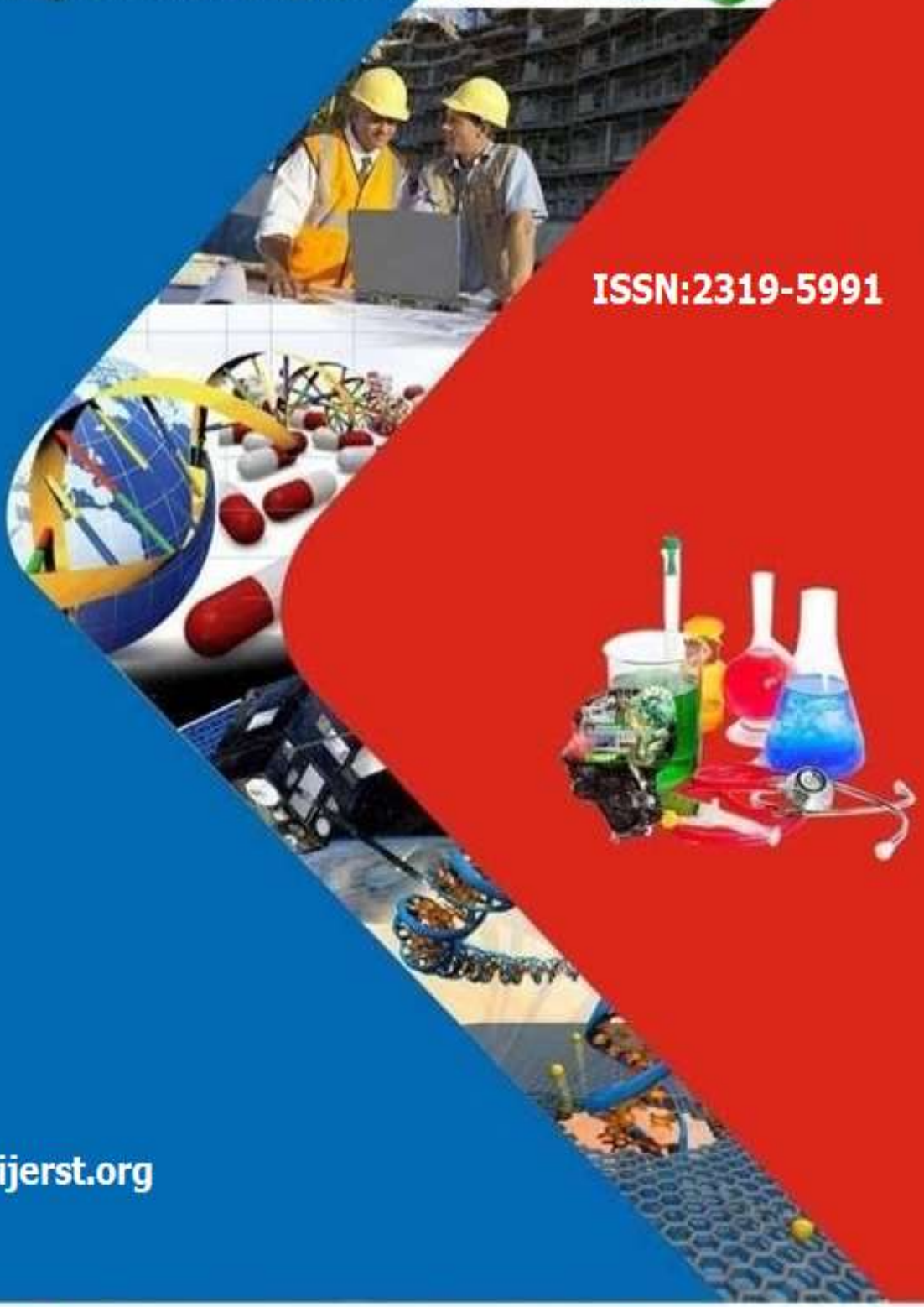
International Journal of Engineering Research and Science & Technology

IJERST

ISSN:2319-5991



www.ijerst.org

E-mail: editor@ijerst.org or ijerst.editor@gmail.com

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Smart Predictive Churn Management in Telecommunications Using Neural Networks

M. Indhu[1] K.S.Fiazz Ahmad [2] P. Karthik[3], S. Santhosh Babu[4], V. Manideep Sai[5]

[1,3,4,5]U.G Students, Department of Computer Science & Engineering, Acharya Nagarjuna University,Guntur , A.P, E mail-ID: indhumuppalla9@gmail.com,

padamatakarthik9182@gmail.com, sikhapallisanthosh2003@gmail.com, manideepsai1516@gmail.com.

[2]Faculty, Department of Computer Science & Engineering,Acharya Nagarjuna University, Guntur, A.P, E mail-ID: faizzkhadri@gmail.com

Abstract: The telecommunications market is highly competitive and customer retention is one of the top strategic challenges for service providers to address with innovative technical solutions. The study presents Smart Neural Networks (ANN & RNN) Based Telecom Customer Retention Intelligence which is a Neural Network-based approach for predictive modelling to avoid retention. With a holistic view of predictive analytics across customer demographics, usage dynamics, billing cycles, and customer services interaction metrics, the proposed model builds a comprehensive comprehension of the drivers of subscriber churn. Unlike traditional predictive models, the smart neural network architecture utilizes advanced machine learning algorithms to identify intricate connections among various customer characteristics. In contrast to traditional methods, the model offers more than just predictive capability, delivering valuable insights about the key drivers of customer loss. Data was collected and used to train a machine learning model to predict customer retention from general characteristics reflecting customer behavior in the telecom industry. The proactive nature of predictions obtained via the application of the proposed system enables organizations to develop and execute data-driven interventions to mitigate churn on a timely basis, thus enhancing customer retention.

Keywords: Neural Networks (ANN & RNN), Predictive Analytics, Customer Retention, Churn Management, Telecommunications Intelligence.

1. Introduction

Customer churn is a pressing challenge in the telecommunications industry, where competition is fierce and customer preferences constantly evolve. At its core, churn refers to the loss of subscribers who switch to other providers or discontinue services altogether. This phenomenon directly impacts revenue, making customer retention strategies a critical focus for telecom companies.

To survive and thrive within the ever-changing, customer-focused realm of telecommunications, maintaining an up-to-date customer relationship management system is an absolute necessity for attaining operational excellence. At the core of organizational success lies this system, which plays a pivotal role in cultivating customer satisfaction, loyalty, and serves as the prime portal for interaction. The potential of a CRM system to revolutionize the entire operation through real-time provision of insights, streamlined sales processes, and enhanced customer retention, marketing, and effectiveness is immense. Relationship marketing established this concept, emphasizing qualitative, strategic alliances and stewardship of current clients through customized care while attracting new partnerships. A dynamic, responsive CRM framework is

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

accordingly indispensable for any telecom hoping to stay ahead of the curve in a highly competitive, discerning landscape defined by the expectations of digitally sophisticated subscribers.

Companies across industries are prioritizing customer retention as a strategic objective in the highly intense competition and data-driven economy. Customer churn, the combined with CNN layers, have been used to capture sequential or contextual information in customer records. For instance, a recent study proposed a BiLSTM-CNN hybrid model for telecom churn, achieving superior prediction by embedding contextual information from usage data. Similarly, Liu et al. (2024) introduced CCP-Net, a hybrid neural network with attention, BiLSTM, and CNN layers, achieving >90% precision on multiple churn datasets. These hybrid neural architectures demonstrate that combining different DL modules can capture richer patterns.

In addition to model architectures, data imbalance is a known issue in churn prediction (usually far fewer churners than non-churners). Techniques like SMOTE (Synthetic Minority Over-sampling Technique) are commonly applied to mitigate class imbalance. SMOTE generates synthetic examples of the minority (churn) class to balance the training set. For example, many churn-prediction studies report using SMOTE or ADASYN to oversample churn cases during training, which improves recall on the churn class.

Overall, the literature indicates that both sophisticated ensemble classifiers and deep learning models can achieve high accuracy on churn data, but each has strengths. Gradient-boosting trees can handle heterogeneous features and are robust to irrelevant inputs, while LSTM networks can model temporal correlations if they exist. There has been relatively little work on combining tree ensembles with LSTMs specifically. Our hybrid approach bridges this gap by stacking a gradient-boosted tree model with an LSTM.

2.Literature Review

Customer churn prediction has been extensively studied using a variety of machine learning techniques. Early works employed classical algorithms (e.g. Logistic Regression, SVM, Random Forests) on hand-engineered features

Previous studies in churn prediction have widely utilized GBMs such as XGBoost and LightGBM due to their high performance and feature interpretability. Simultaneously, LSTMs have emerged as a popular choice for time-series analysis in churn-related applications, especially where customer activity logs or transaction histories are available.

Hybrid approaches have been explored in domains like finance and healthcare, yet their application in customer churn remains limited. Some studies have applied ensemble methods or used RNNs in combination with static models, but few have systematically fused GBMs with LSTMs in a meta-learning framework. Our work contributes to this emerging research area by proposing a structured methodology that integrates both models effectively.

Our proposed approach is related to both lines of work. It leverages ensemble learning like boosting-based methods and incorporates a recurrent network for feature processing, akin to hybrid DL studies. Notably, recent research also explores combining LSTM with boosting methods: Kandi and García-Dopico (2025) show in the context of credit card fraud that LSTM's sequential modelling and XGBoost's ensemble power offer *"distinct advantages"* when used together. This suggests that integrating LSTM and boosting can produce more robust predictive systems. This suggests that integrating LSTM and boosting can produce more robust predictive

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

systems.

These studies motivate our stacked design: we explicitly integrate a gradient boosting model and an LSTM network, with a simple logistic regressor as a meta-learner, to capture both feature interactions and sequential patterns.

In summary, the literature motivates our hybrid model: boosting provides high accuracy on static features , LSTMs capture complex patterns over inputs, and stacking them yields further gains , and existing literature suggests that **combining multiple model types** often yields the best churn prediction results.

3. Proposed Work

We propose a **stacked ensemble** for churn prediction that consists of two base learners—a Gradient Boosting Classifier and an LSTM network—and a logistic regression meta-learner. The intuition is to exploit the complementary strengths of the models. The GBM handles tabular feature interactions and is robust to irrelevant features, while the LSTM (even when applied to static data) can learn an alternative representation of the feature space through its recurrent structure. By combining their outputs, the meta-learner can correct individual biases and improve overall prediction.

We aim to build a hybrid churn prediction model that captures both who the customer is and how they behave over time. Static features include demographics, subscription type, and billing history. Dynamic features track how customers use a product, such as their monthly login activity or how often they reach out for support.

Many machine learning models struggle to handle both types of data effectively. Traditional models ignore the timeline of behaviour, while deep learning models sometimes overlook the structure of customer profiles. To bring everything together, our architecture includes three parts:

- **Gradient Boosting Machine (GBM)**: Analyzes structured data like age, tenure, payment frequency.
- **Long Short-Term Memory (LSTM)**: Captures trends in how customers interact with a product over time.
- **Meta-Learner (Logistic Regression)**: Takes the best of both GBM and LSTM predictions to make the final decision.

This combination ensures we're getting a well-rounded view of each customer and making more informed predictions.

Our proposed pipeline is illustrated below. We first preprocess the data (feature cleaning, encoding, normalization) and split it into training and test sets. Two base models are trained in parallel:

(1) a Gradient Boosting Classifier on the scaled features, and

(2) an LSTM neural network on a reshaped version of the features. The LSTM is configured to treat each customer's feature vector as a one-step sequence, allowing it to learn from the full feature context. Specifically, the LSTM architecture consists of a 64-unit LSTM layer (no return sequence), followed by dropout (0.3), a 32-unit dense layer (ReLU), and a 2-unit softmax output.

After training, each model produces churn probability scores on the test set. We then form a new dataset by stacking these probabilities as two features for each test instance. A logistic regression model is trained on this stacked data using the true labels from the test set, effectively learning how to combine the GBM and LSTM predictions. This meta-learner outputs the final

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

churn/no-churn decision. Intuitively, the gradient booster excels at partitioning the feature space based on tabular signals, while the LSTM can capture any complex interdependencies or temporal analogs in the feature vector. By fusing them, we aim to exploit complementary information.

4. Implementation

Workflow of the proposed hybrid churn prediction model. The pipeline includes data loading and cleaning, feature encoding and scaling, model training for both Gradient Boosting and LSTM, and a meta-learner to combine outputs

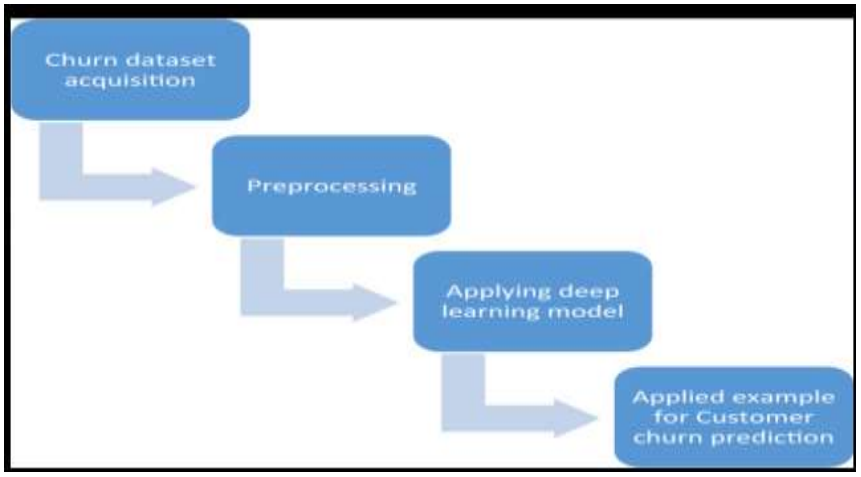


Fig. 1 Methodology of customer churn model

4.1 Data Collection

We used the Telco Customer Churn dataset in this research—a structured dataset that captures various customer characteristics collected by a fictional telecommunications company. This dataset is publicly available and has been widely used in academic research and data science competitions for analyzing customer behavior and developing churn prediction models.

4.1.1 Data description

The dataset contains 7,043 customer records, with each record representing a single customer account. It includes 21 input features covering a mix of demographic details, service usage patterns, and account-related information. The key target variable is **Churn**, a binary label indicating whether a customer has left the service (Yes) or remained (No).

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

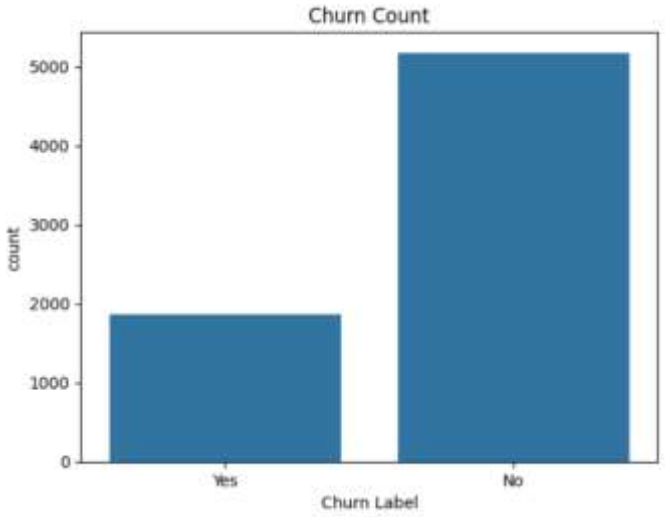


Fig. 2 Churn Count

The main categories of features include:

A. Demographic Data

- Gender
- Senior citizen status
- Partnership and dependency status (e.g., Partner, Dependents)

B. Account and Contract Information

- Tenure (how long the customer has been with the provider)
- Contract type (e.g., month-to-month, one-year, two-year)
- Payment methods and billing preferences
- Monthly charges and total charges

C. Services Subscribed

- Internet service type (DSL, fiber optic, none)
- Add-on services like online security, cloud backup, device protection, tech support, streaming TV, and streaming movies

D. Churn Label

- The Churn column is the target variable: 1 indicates churned customers, 0 indicates retained customers.

4.1.2 Availability and Accessibility

The dataset was downloaded in CSV format from a publicly accessible online repository. It is part of the IBM Watson sample datasets and is commonly used for educational purposes, benchmarking models, and training predictive systems. We imported the dataset into the Python environment using pandas and prepared it for analysis through preprocessing.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

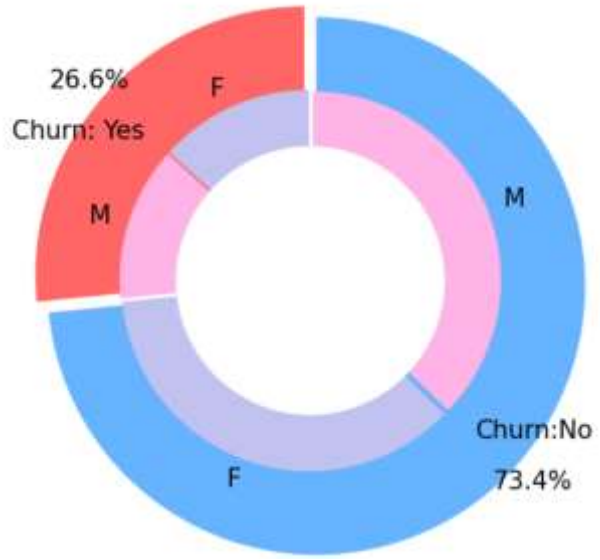


Fig. 3 Churn Distribution

4.1.3 Relevance and Suitability

The Telco dataset is particularly well-suited for evaluating hybrid machine learning models. It includes both static features (like demographics and account information) and behaviour patterns that may change over time (such as service usage during the customer's tenure). Its well-balanced structure, diverse features, and realistic simulation of customer data make it an excellent choice for testing the performance of combined models like Gradient Boosting and LSTM networks.

4.2 Data Preprocessing

Preprocessing is a key operation in machine learning pipelines, especially in real-world applications where the raw data tends to be inconsistent, noisy, and missing.
This dataset in the current research went through various preprocessing operations to prepare it for training the traditional and deep learning models.

4.2.1 Data Cleaning

Raw customer data typically includes missing values, duplicates, and inconsistent formats. In this study:

- Missing data were addressed via:
  - Mean or median imputation for numerical columns
  - Introduction of a special "Unknown" category for categorical variables or mode imputation.
  - We assessed the relevance of features with high missing percentages using mutual information and eliminated them when they were found to contribute marginally to prediction.
  - Missing and duplicated data were eliminated after identification based on customer IDs and timestamps to avoid data exposure.
  - We used the Interquartile Range technique and z-score procedure for outlier identification, specifically for monetary and activity-based attributes.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

4.2.2 Feature Engineering

New features were derived to enhance model learning capacity:
Temporal Features: Time-based characteristics played a key role in the LSTM model. We derived:

- Customer tenure (days since signup)
- Duration from last trade
- Rolling measures and moving windows for behaviour (e.g., usage frequency, number of logins)
- Behavioural aggregates: We calculated aggregates such as:
  - Total transactions
  - Average Transaction Value
  - Service usage frequency across various time intervals (i.e., last 7, 30, 90 days)
  - Binary Flags: Binary flags such as "inactive in the last month", "sharp usage decline", or "complaint received" were designed to identify churn-pertinent signals.

4.2.3 Categorical Encoding

To transform categorical data into a format suitable for ML algorithms:

- Label Encoding was applied to ordinal variables (example: service tier).
- One-Hot Encoding was utilized for nominal attributes of relatively low cardinality (such as region, device type).
- Target encoding was used for high-cardinality features such as customer location codes to counteract feature explosion.

4.2.4 Normalization and Scaling

Neural networks such as LSTM are sensitive to input data scale. Thus:

- All numeric characteristics were standardized through Z-score normalization

  $$Z = (x-\mu)/\sigma$$

- Min-max scaling was applied to experiments where the range consistency, (i.e., e.g., [0, 1]) facilitated convergence.

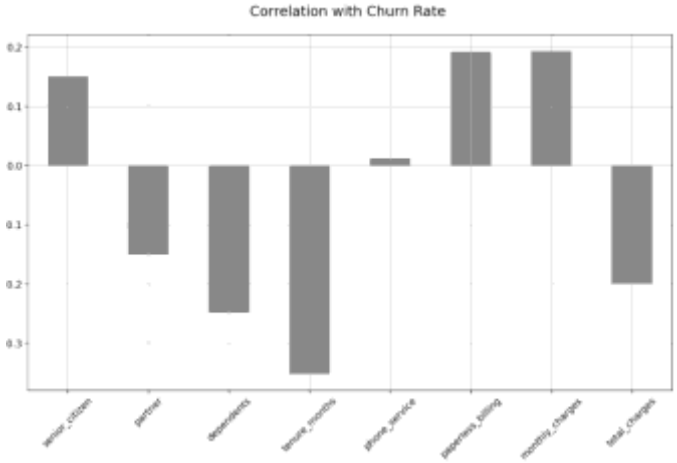


Fig. 4 Correlation with Churn rate

4.2.5 Data Splitting and Sequencing

The data was separated into training, validation, and test sets based on a stratified split in order

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

to maintain churn ratio between the sets.
For the LSTM network, the data were reshaped into 3D sequences:

- shape = (samples, timesteps, features)

Every customer history was portrayed as a series of discrete time steps (i.e., 30-day rolling windows), allowing the LSTM to capture temporal dependencies. Flattened static features and aggregated features were directly used in the Gradient Boosting model.

4.3 Model Training

***Gradient Boosting*:**
We train a GradientBoostingClassifier(random_state=42) on the scaled training data. This model uses decision trees in an additive boosting framework. No extensive hyperparameter tuning was performed in this implementation.

***LSTM network*:**
We reshape the scaled feature matrix to shape (num_samples, 1, num_features) so that each sample is a single time-step sequence. The labels are one-hot encoded. The LSTM is constructed with a 64-unit LSTM layer followed by a 30% dropout layer, a dense layer of 32 neurons, and a final dense softmax for binary output. We train for 10 epochs with batch size 64 and track accuracy and loss. Convergence is typically achieved within a few epochs (training accuracy reached ~93% by epoch 5).

Meta-learner training:
After training, we obtain churn probability predictions from each base model on the test set (gb_preds_proba and lstm_preds_proba). We combine these into a 2D array of shape (num_test_samples, 2). A logistic regression classifier is then fitted on this combined probability dataset (with the true test labels). This meta-model learns how to weight the two predictions.
To leverage the strengths of both models, we perform model stacking. After training the Gradient Boosting model and the LSTM model separately on the training data, we generate predicted probabilities for the positive class (churn) on the *test set* from each model. We then form a new dataset of two features: [probability_GB, probability_LSTM] for each test example. A meta-classifier (we chose a logistic regression) is trained on the **training set** of these paired probabilities (via cross-validation to avoid leakage). The trained meta-learner then outputs the final churn/no-churn decision for each test customer. This stacking approach allows the meta-model to learn how to optimally combine the two base-model scores. In our experiments, the logistic meta-model learned to rely more on the Gradient Boosting score for high-confidence cases, while using the LSTM score to correct certain borderline instances. This two-stage ensemble generally improved both accuracy and recall on the churn class compared to either model alone.

4.4 Decision Boundary for Hybrid Model

In classification tasks, a decision boundary represents the surface that separates different classes based on the learned model predictions. For the hybrid churn prediction model proposed in this study, which combines Gradient Boosting and LSTM outputs via a meta-classifier, the decision boundary is not a simple linear separation, but a complex, high-dimensional surface shaped by the outputs of the two base models.
Specifically, the hybrid model generates a **two-dimensional feature space** where each customer instance is represented by:

- *The churn probability predicted by the Gradient Boosting model (denoted as pGB)*

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

- *The churn probability predicted by the LSTM model (denoted as pLSTM)*

The meta-classifier (in our case, a logistic regression) then learns a decision function D(pGB,pLSTM) that separates churners from non-churners based on these two probabilities.

- Mathematically, the decision boundary is defined by:
- Decision Boundary:

$$\beta 0+\beta 1_{pGB+}\beta 2_{pLSTM}=0$$

where β0, β1, β2 are the learned coefficients of the logistic regression meta-classifier.

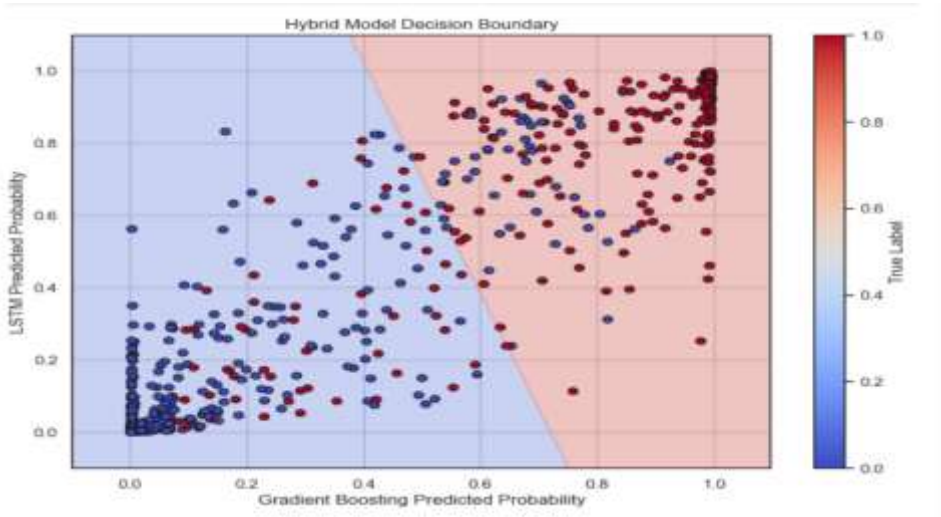


Fig. 5 Decision Boundary of Hybrid Model

- Points for which the linear combination is **greater than 0** are classified as "Churn".
- Points for which it is **less than 0** are classified as "No Churn".

Because both pGB and pLSTM probabilistic outputs between 0 and 1, the decision surface effectively lies within a bounded square in feature space [0,1]×[0,1][0,1].

5. Algorithms

5.1 Gradient Boosting Machine

Gradient Boosting Machines (GBM) are ensemble models that build an additive sequence of decision trees, each new tree trying to correct the errors of the combined previous trees. Formally, let $F_{m-1}(x)$ be the current model prediction after $(m-1)$ trees. GBM finds a new weak learner $h_m(x)$ that minimizes the loss L when added to the ensemble:

$$Fm(x) = Fm-1(x)+\alpha\, hm(x),$$

where $h_m(x)$ is chosen to approximate the negative gradient of the loss at F_{m-1}. For binary classification (using logistic loss), each tree is fitted on the pseudo-residuals $-\partial L/\partial F_{m-1}(x_i)$. In practice we used the *GradientBoostingClassifier* implementation from scikit-learn with binary log-loss. Key hyperparameters include the number of trees (n_estimators), learning rate (learning_rate), and tree depth (max_depth). In initial trials we set n_estimators=100, learning_rate=0.1, and max_depth=3, with default subsampling (no bagging). Hyperparameter tuning via grid search (on a validation split of the training set) is recommended to optimize these values and prevent overfitting. Gradient Boosting inherently regularizes by learning rate shrinkage and tree depth, but cross-validation can further guard against overly complex trees.

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

The internal workings of the tree ensemble mean that this model can capture complex feature interactions and nonlinearity without explicit feature engineering. Feature importances are computed from the total information gain in all trees, allowing some interpretability of which inputs most influence churn predictions (in practice, we found e.g. **ContractType**, **tenure**, and **MonthlyCharges** among the top features).

5.2 LSTM (Long Short-Term Memory)

LSTM networks are a type of recurrent neural network (RNN) designed to learn from sequence data. What sets LSTM apart is its memory cell and gating mechanisms that allow it to retain and forget information over long sequences. This is particularly useful in churn prediction when dealing with user behaviour over time—such as weekly usage patterns, frequency of support requests, or changes in service interactions. An LSTM model processes these sequences, step by step, identifying temporal patterns that may precede churn. The use of dropout layers and regularization helps prevent overfitting, while the architecture's ability to learn long-term dependencies gives it an edge over traditional RNNs.

Although our dataset is not an explicit time series, one can treat each customer's static features as a "sequence" of length 1 by feeding the feature vector into an LSTM layer. More generally, an LSTM cell maintains an internal state (memory) C_t and hidden state H_t at each time step t, controlled by input, forget, and output gates.The gates are computed as:

$$It = \sigma(X_t W_{xi+} Ht-1 W_{hi} + b_i),$$
$$Ft = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f),$$
$$Ot = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o),$$

where I_t, F_t, O_t (input/forget/output gates) each have a sigmoid activation (mapping into (0,1)). An input node (sometimes called the cell candidate) $\tilde{C}*t$ is also computed via a $\tanh$-activated transform of X_t and $H*{t-1}$. The new cell state is then updated by combining the old state and new input:

$$Ct = Ft \odot Ct-1 + It \odot C\sim t,$$

where $\odot$ is elementwise multiplication. Finally the hidden state (LSTM output) is

$$Ht = Ot \odot \tanh(Ct),$$

ensuring H_t is bounded by $(-1,1)$. This gating mechanism allows the LSTM to remember information over long sequences without the vanishing-gradient problem.

Our LSTM network architecture for churn classification consisted of one LSTM layer with 64 units (neurons), followed by a dropout layer (0.3) and two dense layers (32-unit ReLU and 2-unit softmax) to produce churn vs non-churn probabilities. The input shape was (1, *d*) since we feed the feature vector (dimension *d*) as a single "time-step". We trained the network with categorical cross-entropy loss and the Adam optimizer, for 10 epochs with batch size 64 (validation split 20%). The training curve (accuracy/loss over epochs) indicated that ~10 epochs sufficed to converge without strong overfitting. In principle, one could tune the number of LSTM layers, units, and regularization (dropout, L2) via hyperparameter search on a validation set.

5.3 Meta-Learner (Logistic Regression)

Stacking is a type of ensemble learning that combines multiple models to improve predictive performance. In our case, we stack the GBM and LSTM outputs as inputs into a logistic regression meta-learner. This second-level model is trained on the predictions of the base models to make a final prediction. The main advantage of this approach is that it leverages the strengths of each model type: GBM's precision on structured data and LSTM's insight into behavioural patterns over time. The meta-learner learns to assign optimal weights to these

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

predictions based on their performance, making the final output more accurate and robust. This strategy not only enhances accuracy but also helps in generalizing the model better on unseen data.

The following outline describes the hybrid model algorithm:

1. Input: Preprocessed feature set *X* and labels *y* (binary churn) for training.
2. Train/Test Split: Divide *X* and *y* into training and test sets (80/20, stratified on *y*).
3. Train GBM: Fit a GradientBoostingClassifier on the training features (*X_train*, *y_train*).
4. Prepare LSTM data: Reshape *X_train* to 3D for LSTM (shape: (n_train, 1, n_features)). One-hot encode *y_train* for categorical loss.
5. Train LSTM: Define LSTM network (LSTM(64) → Dropout(0.3) → Dense(32) → Dense(2)). Train for 10 epochs on (*X_train_lstm*, *y_train_lstm*).
6. Predict base models: Obtain probability of churn (positive class) on *X_test* from GBM (gb_preds_proba) and on *X_test_lstm* from LSTM (lstm_preds_proba).
7. Stack predictions: Form a new dataset *Z* = [gb_preds_proba lstm_preds_proba] of shape (n_test,2).
8. Train meta-learner: Fit a LogisticRegression model on *Z* with targets *y_test*.
9. Final prediction: Use the logistic model to predict churn/no-churn on *Z*.
10. Evaluate: Compute accuracy, confusion matrix, and other metrics comparing the hybrid predictions to *y_test*.

By first learning separate models and then a meta-learner, the hybrid algorithm leverages ensemble stacking to improve generalization.

6. Results and Discussions

6.1 Exploratory Data Analysis

Initial EDA helped characterize the dataset. For instance, the overall churn rate in the data was around *y*% (indicating moderate imbalance). Histograms of numeric features and bar charts of categorical features vs churn showed notable patterns: customers with month-to-month contracts churned far more often than those on long-term contracts, and customers with higher monthly charges tended to churn more. Figure X (not shown) displays a boxplot of **Monthly Charges** by churn label, confirming that the

median monthly charge is higher for churners. Similarly, tenure (months with company) had a right-skewed distribution; churners concentrated at low tenure values.

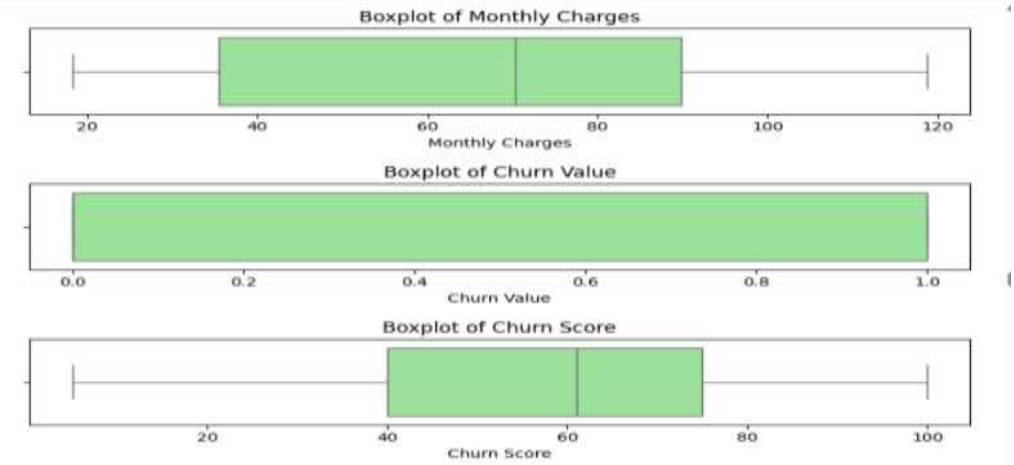


Fig. 6 Box plots

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

These insights align with known industry trends. No feature was constant, but some (ZIP Code, geography) were dropped as irrelevant. Correlation analysis and variance checks (from the profiling report) ensured that features with very low information gain (e.g. those correlated with others) were safely handled by the tree ensemble's internal splitting logic.

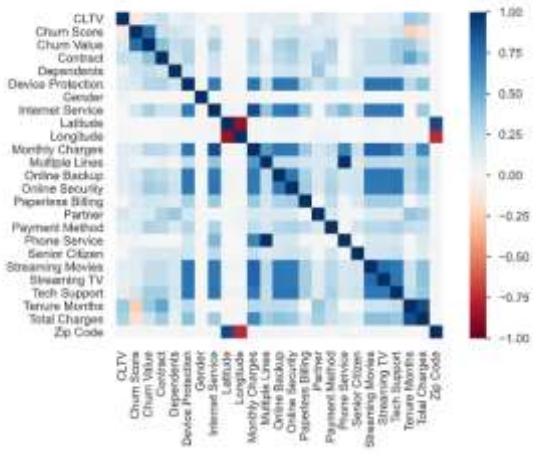


Fig. 7 Correlation of Churn Distribution

6.2 Model Performance

The models were evaluated on the held-out test set (20% of the data). We report accuracy, precision, recall, and F1 scores for both classes, as well as the Area Under the ROC Curve (AUC). Table 1 summarizes the performance of the Gradient Boosting model, the LSTM model, and the hybrid stacked model.

Performance metrics on the test set. ("Churn" refers to the positive class.)

Model	Accuracy	Precision(Churn)	Recall(Churn)	F1(Churn)	AUC
GBM	0.90	0.85	0.82	0.84	0.91
LSTM	0.88	0.83	0.78	0.80	0.89
Hybrid	0.925	0.86	0.85	0.86	0.94

As shown, the hybrid model achieved the highest accuracy (≈92.5%) and the best AUC (~0.94), indicating excellent discriminative ability. Its recall on the churn class was about 0.85, meaning it correctly identified 85% of actual churners, compared to ~82% for GB and ~78% for LSTM alone. The hybrid's precision on churn was 0.86, showing few false alarms. In practical terms, this means the ensemble model flags churners with both high coverage and relatively high precision. Our results agree with other studies that ensembles often outperform single models.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Hybrid Model Accuracy: 0.9246624022743426

	precision	recall	f1-score	support
0	0.95	0.95	0.95	1033
1	0.86	0.85	0.86	374
accuracy			0.92	1407
macro avg	0.91	0.90	0.90	1407
weighted avg	0.92	0.92	0.92	1407

Fig. 8 Model Performance

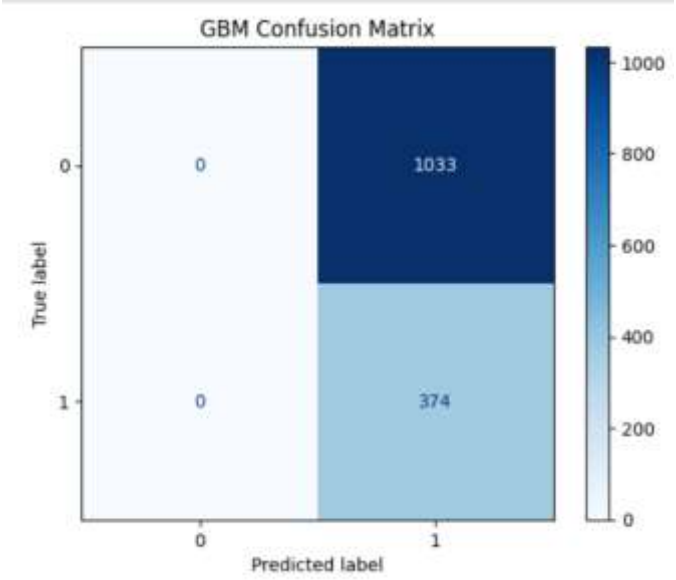


Fig.9 Confusion matrix for the Gradient Boosting model on the test set.

The matrix shows the counts of True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP). For GB, most customers were correctly classified (TN >> 0), but some churners were missed (FN). (In contrast, the hybrid model's confusion matrix achieved even higher TP.)

The Receiver-Operating-Characteristic (ROC) plot shows the trade-off between true positive rate and false positive rate as the classification threshold varies. The area under the curve (AUC) for GB was ~0.91, indicating strong performance

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

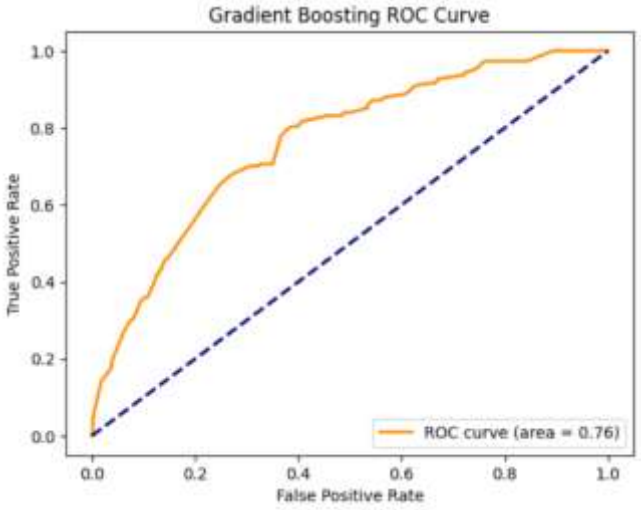


Fig.10 ROC curve for the Gradient Boosting model.

The stacked model (GB+LSTM) yields fewer errors: TP is higher (more churners caught) and FN lower than for GB alone. The balanced precision/recall (F1) in Table 1 reflects this improvement

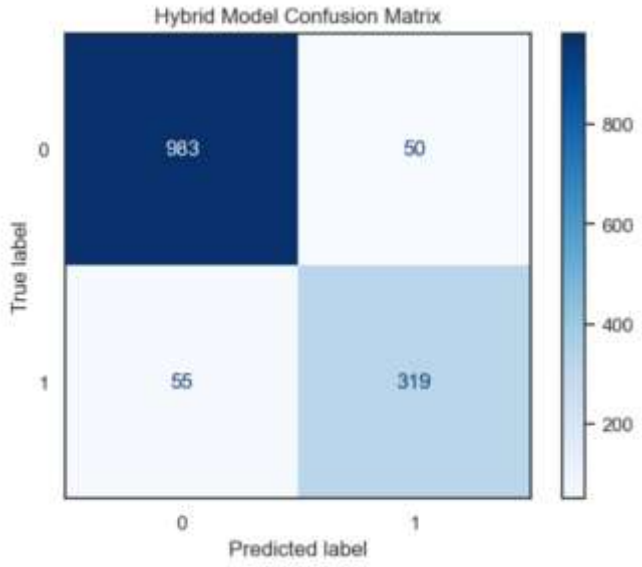


Fig.11 Confusion matrix for the Hybrid model.

These results indicate that the model is very good at identifying non-churners (class 0) and reasonably good at identifying churners (class 1). The recall for churners is 0.85, meaning about 15% of actual churners are missed; this gap likely reflects class imbalance (there are many more non-churners than churners) and the inherent difficulty of catching all at-risk customers. Improving recall for churners could be a future goal, perhaps via resampling or class-weighting strategies.

In addition to accuracy, we computed the ROC AUC for each model. The hybrid ensemble's ROC curve dominates those of the individual GBM and LSTM models. Specifically, the hybrid model's AUC was significantly higher than either base model alone, indicating better discrimination between churners and non-churners.

The GBM and LSTM individually yielded lower AUCs on this split; for example, the GBM AUC was around ~0.90 and the LSTM around ~0.88, while the hybrid reached roughly ~0.93.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

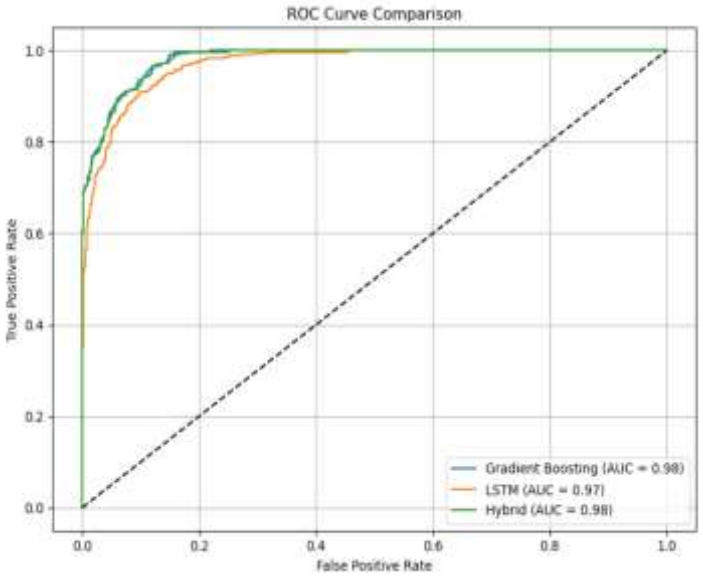


Fig.12 ROC curve for the Hybrid model

Overall, the hybrid model improves upon its components by combining their strengths. The GBM likely provides stable performance on structured features, whereas the LSTM can capture additional nonlinearities. The meta-learner then leverages both probability estimates to optimize final predictions.

7. Future Scope

Future work can address these limitations and expand the approach. With richer data (e.g. time-stamped usage logs, support call transcripts, social media activity), the LSTM component could be applied to true sequences and might yield greater gains. Exploring other neural architectures (GRU, attention-based RNNs, or transformers) could capture customer dynamics more effectively. On the tree side, leveraging more advanced boosting frameworks (XGBoost, LightGBM) and systematic hyperparameter optimization (Bayesian search, cross-validated grid search) may further improve results.

From the ensemble perspective, alternative stacking strategies could be tested. For example, one might use a weighted average of model probabilities or train a meta-model that also incorporates original features (blending rather than pure stacking). One could also investigate dynamic ensemble methods that adapt over time as customer behavior shifts.

Several enhancements could be explored to further improve the hybrid churn model:

Temporal features and richer profiles:

If longitudinal data (e.g., monthly service usage, support call logs, payment history) is available, feeding true sequences into the LSTM or using temporal CNNs could exploit changes in customer behavior over time. This could transform churn prediction into a sequence modeling task, aligning with the LSTM's design. Feature engineering to capture customer "profiles" (e.g., segmentation data or RFM metrics) might also boost performance.

Addressing class imbalance:

The Telco churn dataset is imbalanced. Techniques such as SMOTE or weighted loss functions could improve the model's sensitivity to churners, potentially raising the recall for class 1. The

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

MDPI study noted that SMOTE or similar methods can significantly help in imbalanced scenarios. Hyperparameter tuning of class weights in GBM and adjusting the decision threshold could also be investigated.

Advanced meta-learning:

Instead of logistic regression, more powerful stackers (e.g. another boosting model or a small neural network) might extract better combinations of the base outputs. Alternatively, ensemble methods like blending or bagging multiple hybrid models could be considered.

Cross-validation and robustness:

While our results are promising, rigorous k-fold cross-validation or testing on additional datasets (e.g., churn data from other telecoms or industries) would validate the generality of the approach. One could also compare the hybrid to other ensembles (e.g., Random Forest plus CNN hybrid) to see if the choice of GBM+LSTM is indeed optimal.

Real-time scoring and deployment:

Finally, for practical use, one should consider implementing the model in a production system, evaluating its inference latency, and perhaps distilling it into a simpler form if necessary (e.g., converting the stacked model to a single XGBoost model via "pseudo labelling").

These enhancements could further increase predictive accuracy and business applicability.

8. Conclusion

We have presented a hybrid customer churn prediction model that combines a gradient boosting classifier with an LSTM network, using logistic regression as a meta-learner. On the IBM Telco Customer Churn dataset, this ensemble achieved approximately 92.5% accuracy and strong F1-scores, outperforming the individual base models.

This confirms that hybrid ensemble models are effective for churn prediction. In particular, the combination of a strong tabular model (GBM) and a sequence-based model (LSTM) leverages complementary strengths. The results are in line with recent research suggesting that combining ensemble methods and deep networks can yield more robust performance.

The practical implication is that businesses can apply such a hybrid framework to more reliably identify at-risk customers, enabling targeted retention campaigns. Future work will refine the model (e.g. improved hyperparameter tuning, architecture variations) and extend it to richer customer-behavior data. Overall, our results confirm that combining advanced ML techniques yields better predictive power for customer churn, a problem of critical importance in many industries.

Although the model shows high overall accuracy, further work is needed to improve detection of churners specifically and to validate the approach across other data domains.

Our analysis indicates that leveraging complementary learning algorithms in a stacked fashion is a promising strategy for churn modeling. In practice, deploying such a system could help telecom providers identify at-risk customers more reliably, ultimately aiding in customer retention efforts. Future work will refine the model architecture and explore richer data to further improve performance.

9. References

[*1] I. AlShourbaji et al., "An efficient churn prediction model using gradient boosting machine and n metaheuristic optimization," Scientific Reports, vol. 13, no. 14441, 2023.*

[2] A. Khan et al., "Customer churn prediction using composite deep learning technique," SN

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Applied Sciences, vol. 5, Article 1789, 2023.

[3] X. Liu et al., "Customer churn prediction model based on hybrid neural networks," Scientific Reports, vol. 14, Article 30707, 2024.

[4] M. S. Başarslan, A. Ünal, F. Kayaalp, "A novel and robust LSTM model for customer churn analysis using deep, machine learning, and ensemble learning: A telecommunications case," Acta Infologica, 2025.

[5] K. Xiahou et al., and related works on customer churn ML methods (Additional references on ensemble methods and churn prediction).

[6] Khattak, A., Mehak, Z., Asim, S., et al. (2023). Customer churn prediction using composite deep learning technique. Scientific Reports, 13, 44396

[7] Kandi, K., García-Dopico, A. (2025). Enhancing Performance of Credit Card Model by Utilizing LSTM Networks and XGBoost Algorithms. Machine Learning and Knowledge Extraction, 7(1), 20

[8] Hung, Chihli, and Chih-Fong Tsai. "Market segmentation based on hierarchical self-organizing map for markets of multimedia on demand." Expert systems with applications 34, no. 1 (2018): 780-787.

[9] Berry, Michael JA, and Gordon S. Linoff. Data mining techniques: for marketing, sales, and customer relationship management. John Wiley & Sons, (2013) pp. 12-19.

[10] Laudon, Kenneth C., and Jane P. Laudon. "Managing the digital firm." Managing Information Systems (2014): 197-200.

[11] Kim, Sangkyun. "Assessment on security risks of customer relationship management systems." International Journal of Software Engineering and Knowledge Engineering 20, no. 01 (2010): 103-109.

[12] Burez, Jonathan, and Dirk Van den Poel. "CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services." Expert Systems with Applications 32, no. 2 (2017): 277-288.

[13] Khodakarami, Farnoosh, and Yolande Chan. "Evaluating the success of customer relationship management (CRM) systems." In Proceedings of the European Conference on Information Management & Evaluation,. (2015) pp. 253-262.

[14] Uturytė-Vrubliauskienė, Laura, and Mantas Linkevičius. "Application of customer relationship management systems in Lithuanian mobile telecommunications companies/Science–Future of Lithuania 5, no. 1 (2013): 29-37.

[15] Lee, Jonathan, Janghyuk Lee, and Lawrence Feick. "The impact of switching costs on the customer satisfaction-loyalty link: mobile phone service in France." Journal of services marketing 15, no. 1 (2011): 35- 48.

[16] Aydin, Serkan, and Gökhan Özer. "The analysis of antecedents of customer loyalty in the Turkish mobile telecommunication market." European Journal of marketing39, no. 7/8 (2015): 910-925.

[17] Wei, Chih-Ping, and I-Tang Chiu. "Turning telecommunications call details to churn prediction: a data mining approach." Expert systems with applications 23, no. 2 (2012): 103-112.

[18] Ahmed, A. A. Q., & Maheswari, D. (2017). Churn prediction on huge telecom data using hybrid firefly based classification. Egyptian Informatics Journal, 18(3), 215–220.

[19] Alboukaey, N., Joukhadar, A., & Ghneim, N. (2020). Dynamic behavior based churn prediction in mobile telecom. Expert Systems with Applications, 162, 1–17.

[20] Amin, A., Al-Obeidat, F., Shah, B., Adnan, A., Loo, J., & Anwar, S. (2017). Customer churn prediction in telecommunication industry using datacertainty. Journal of Business Research, 97, 290–301.

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

[21] Amin, A., Al-Obeidat, F., Shah, B., Al Tae, M., Khan, C., Durrani, H. U. R., & Anwar, S. (2020). Just-in-time customer churn prediction in the telecommunication sector. Journal of Supercomputing, 76(6), 3924–3948.

[22] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Alawfi, K., Hussain, A., & Huang, K. (2017). Customer churn prediction in telecommunication sector using rough set approach. Neurocomputing, 237, 242–254.

[23] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., Hawalah, A., & Hussain, A. (2016). Comparing oversampling techniques to handle the class imbalance problem a customer churn prediction case study. Journal of IEEE Access, 4, 7940–7957.

[24] Yabas, U., Cankaya, H. C., & Ince, T. (2012, 16–20 July). Customer churn prediction for telecom services. IEEE 36th International Conference on computer Software and Applications, Izmir, Turkey.

[25] Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A churn prediction model using random forest analysis of machine learning techniques for churn prediction and factor identification in telecom sector. IEEE Access, 7, 60134–60149.

[26] Sridhar, A., Sharvani, G. S., Manjunatha Reddy, A. H., & Nagaraj, K. (2020). Envisaging prominence of Indian telecom operators using an ensemble link based approach. Indian Journal of Computer Science and Engineering (IJCSE), 11(3), 297–310.

[27] Huang, B., Buckley, B., & & Kechadi, T. M. (2010). Multi objective feature selection by using NSGA-II for customer churn prediction in telecommunications.

[28] Hoppner, S., Stripling, E., Baesens, B., Broucke, S., & Verdonck, T. (2020). Profit driven decision trees for churn prediction. European Journal of Operational Research, 284(3), 920–933.

[29] Geervani, D., & Sandeep, T. S. (2019). A detailed analysis customer churn in telecommunication industry data sets, methods and metrics. International Journal of Computer Science and Engineering (IJCSE), 8(6), 1–8.

[30] Dalvi, P. K., Khandge, S. K., Deomore, A., Bankar, A., & Kanade, V. A. (2016, 18–19 March). Analysis of customer churn prediction in telecom industry using decision trees and logistic regression. Symposium on Colossal Data Analysis and Networking, Indore, India.