Enhancing Customer Churn Analysis by Using Real-Time Machine Learning Model

Haitham Ghallab, Mona Nasr, Hanan Fahmy
Department of Information Systems-Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt

Abstract*—Customer churn, the loss of customers to competitors, poses a significant challenge for businesses, particularly in competitive industries such as banking and telecommunications. As a result, several customer churn analysis models have been proposed to identify at-risk customers and enable top managers to implement strategic decisions to mitigate churn and improve customer retention. Although the existing models provide top managers with promising insights for churn prediction, they rely on a batch-based training approach using fixed datasets collected at periodic intervals. While this training approach enables existing models to perform well in relatively stable environments, they, unfortunately, struggle to adapt to dynamic settings, where customer preferences shift rapidly, especially in industries with volatile market conditions, such as banking and telecom. Where, in dynamic environments, data distribution can change significantly over short periods, disabling existing models to maintain efficiency and leading to poor predictive performance, increased misclassification rates, and suboptimal decision-making by top executives, ultimately exacerbating customer churn. To address these limitations, this research proposes RCE, a real-time, continual learning-based, ensemble learning model. RCE integrates an event-driven development approach for real-time churn analysis with a replay-based continual learning mechanism to adapt to evolving customer behaviors without catastrophic forgetting, and RCE implements a stacked ensemble learning for customer churn classification. Unlike existing models, RCE continuously processes streaming data, ensuring adaptability and generalization in fast-changing environments, and providing instantaneous insights that enable decision-makers to respond swiftly to emerging risks, market fluctuations, and customer behavior changes. RCE is evaluated using the Churn Modelling benchmark dataset for European banks, achieving performance with a 95.65% accuracy; however, in dynamic environments, RCE accomplishes an average accuracy (ACC) of 86.75% and an average forgetting rate (FR) of 13.25% across tasks $\boldsymbol{T_i}$. The results demonstrate that RCE outperforms existing models in predictive accuracy, adaptability, and robustness across multiple tasks, especially in dynamic environments. Finally, this research discusses the proposed model's limitations and outlines directions for future improvements in real-time customer churn analysis.

Keywords*—*Customer churn; real-time analysis; continual learning; machine learning; event-driven development; stacked ensemble learning; replay-based approach

I. Introduction and Problem Definition

Customer churn, also referred to as customer attrition, is the process by which customers discontinue their relationship with a business, often opting for competitors that offer better services or incentives [1]. This phenomenon presents a serious challenge for companies, particularly in industries such as banking and telecommunications, where customer retention directly impacts profitability and long-term sustainability. High churn rates result in significant financial losses, with businesses spending considerably more to acquire new customers than to retain existing ones. For instance, studies indicate that acquiring a new customer can be up to five times more expensive than retaining one [2]. Top managers and decision-makers face tremendous difficulties in addressing customer churn due to the rapidly changing nature of customer preferences, evolving market trends, and competitive pressures. Predicting churn accurately requires timely insights into customer behavior, allowing organizations to develop effective retention strategies and mitigate potential losses. To tackle this challenge, machine learning-based customer churn analysis models have been widely adopted to predict which customers are likely to leave. These models help businesses take proactive measures such as personalized marketing campaigns, loyalty programs, and improved customer service [1, 3].

Despite their utility, existing churn prediction models predominantly rely on a batch-based training approach, where models are trained using fixed datasets collected at periodic intervals (e.g., monthly or quarterly). While this approach enables models to perform well in relatively stable environments, it fails to capture the dynamic nature of industries, where customer preferences shift rapidly. Sectors such as banking and telecommunications are highly volatile, with market conditions and consumer behaviors evolving over short timeframes [4]. In such environments, traditional churn models struggle to maintain predictive accuracy as they become outdated between training cycles. Consequently, decision-makers receive suboptimal insights, leading to incorrect managerial strategies that exacerbate customer churn rather than mitigating it. The inability of these models to generalize to new patterns and adapt to shifting customer preferences results in increased misclassification rates, diminished decision-making accuracy, and significant revenue losses [5].

To address these limitations, this research proposes RCE, a real-time, continual learning-based machine learning model designed to enhance customer churn analysis. RCE leverages an event-driven development approach to enable real-time churn prediction, ensuring that decision-makers receive up-to-date insights as customer behaviors evolve. Additionally, RCE incorporates a replay-based continual learning mechanism, allowing it to adapt to new customer behaviors while mitigating the effects of catastrophic forgetting. Unlike traditional batch-based models, RCE processes an ongoing stream of data, enabling businesses to react swiftly to market fluctuations, cybersecurity threats, and changes in customer preferences. By continuously learning from new data, RCE enhances adaptability and generalization, ensuring robust performance in dynamic environments as shown in Fig. 1.

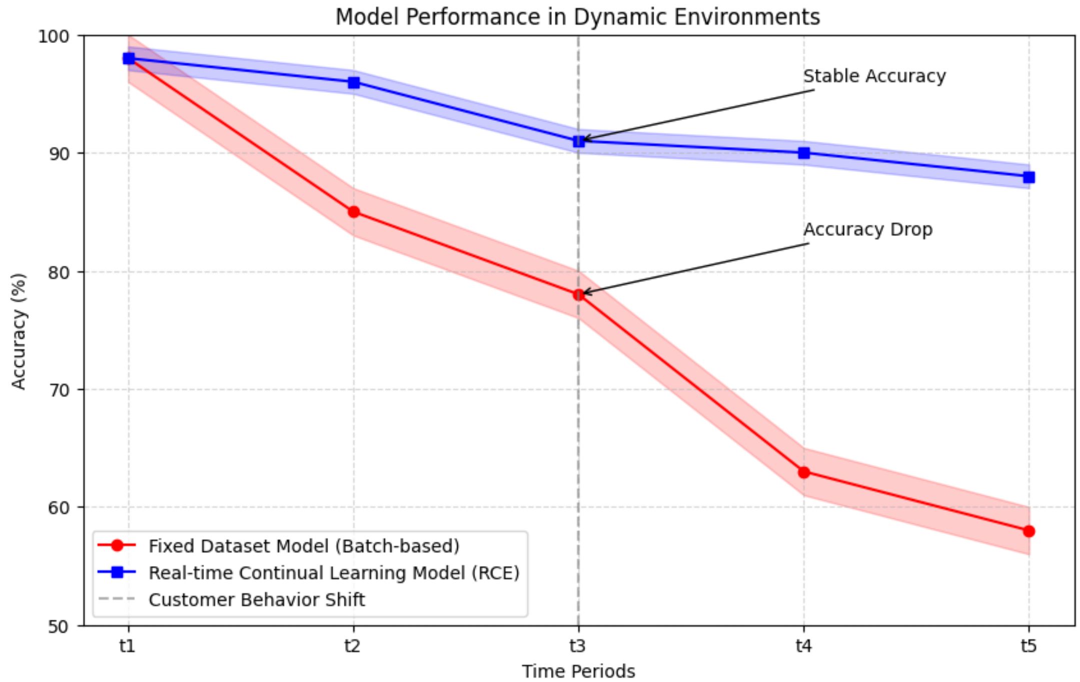

Fig. 1. Batch-based models versus RCE (proposed model) over time, showing RCE's stability in dynamic environments.

The effectiveness of RCE has been evaluated using the Churn Modelling benchmark dataset for European banks [6], where it achieved a performance with a 95.65% accuracy, however in dynamic environments, it accomplishes an average accuracy of 86.75% and an average forgetting rate of 13.25% across tasks T_i. The experimental results demonstrate that RCE outperforms existing models in predictive accuracy, adaptability, and robustness across multiple tasks, particularly in fast-changing environments. By providing instantaneous insights, RCE empowers decision-makers to implement proactive risk management strategies, optimize resource allocation, and reduce costs, ultimately improving customer retention. Additionally, the key features and contributions of this research include:

- Introducing a real-time, continual learning model called RCE designed to overcome the limitations of existing churn analysis approaches.
- Integrating RCE with an event-driven development approach to allow RCE to process an ongoing stream of customer data.
- Implementing RCE using stacked ensemble learning for customer churn classification.
- Applying RCE using a replay-based continual learning technique to enable RCE to adapt to evolving customer behaviors without catastrophic forgetting.
- Enabling RCE to learn continuously from new data, ensuring robust performance in dynamic environments. And qualifying RCE to provide real-time predictions, allowing decision-makers to respond swiftly to market fluctuations, cybersecurity threats, and customer behavior changes.
- Evaluating RCE using the Churn Modelling benchmark dataset for European banks, where it achieved a performance with 95.65% accuracy, however in dynamic environments, it accomplishes an average accuracy of 86.75% and an average forgetting rate of 13.25% across tasks T_i.

The remainder of this study is organized as follows: Section II reviews related work in customer churn analysis and existing machine learning models. Section III presents the background and key concepts of continual learning, stacked ensemble learning, and event-driven data processing. Section IV introduces the RCE model, explaining its architecture and operational mechanism. Section V details the evaluation methodology and experimental results, comparing RCE's performance with conventional models. Finally, Section VI provides the conclusion, discusses research limitations, and outlines future directions for improving real-time customer churn analysis.

II. Related Work

Customer churn prediction has been extensively studied in various industries, with machine learning playing a crucial role in developing accurate forecasting models. As illustrated in Table I. Several recent studies have explored different approaches to customer churn analysis, leveraging ensemble methods, explainability techniques, and novel optimization strategies.

TABLE I. EXISTING CUSTOMER CHURN ANALYSIS MODELS VS RCE PROPOSED MODEL

Model Name	Accuracy (%)	Dataset	Is Adaptable to Ongoing Tasks?	Training Approach
Random Forest [7]	91.66	Telecom CUSTOMER Churn (Maven Analytics)	No	Batch-based
XGBoost [8]	83	Bank Customer Churn (Kaggle)	No	Batch-based
GBM [9]	81	Customer Churn (Kaggle)	No	Batch-based
IDA-HGOAML [10]	94	Telecom Churn (Kaggle)	No	Batch-based
Ensemble-Fusion [11]	95.35	Company Customer Production Line	No	Batch-based
Stacked Model [12]	95.13	Bank Customer Churn (Kaggle)	No	Batch-based
LightGBM [13]	80.70	Telco Customer Churn (Kaggle)	No	Batch-based
CatBoost [14]	81.19	Telecom Customer Churn	No	Batch-based
RCE (Proposed Model)	95.65	Bank Customer Churn (Kaggle)	Yes	Continual-based

V. Chang et al. (2024) investigated the effectiveness of ensemble learning models in predicting customer churn within the telecommunications sector [7]. The study implemented Decision Trees, Boosted Trees, and Random Forests, demonstrating that the latter achieved the highest predictive accuracy of 91.66% using the Telecom Customer Churn dataset from Maven Analytics. The research emphasized the importance of explainable AI (XAI) techniques, such as LIME and SHAP, to provide interpretability, enabling customer relationship managers to make data-driven decisions to mitigate churn [7].

P. P. Singh et al. (2024) analyzed customer churn in the banking industry by applying multiple machine learning algorithms to the Bank Customer Churn Prediction dataset. The study found that XGBoost outperformed other models with an accuracy of 83%, followed by Random Forest at 78.3%. A key contribution of this research was the development of a data visualization RShiny app to assist bank management in understanding churn trends and making informed retention strategies [8].

S. S. Poudel et al. (2024) focused on the interpretability of churn prediction models in the telecommunications industry [9]. The study highlighted the limitations of traditional classification approaches in providing actionable insights for decision-making. By incorporating explainable models such as Gradient Boosting Machine (GBM) alongside SHAP and scatter plots, the research improved transparency. GBM achieved an accuracy of 81% using a Kaggle customer churn dataset, and a Wilcoxon signed rank test validated its superior performance over other models [9].

E. Akhmetshin et al. (2024) introduced a novel IDA-HGOAML model that combines multiple machine learning techniques for customer churn prediction. The proposed method integrated data preprocessing, feature selection, and hyperparameter tuning to enhance classification performance. The model achieved a 94% accuracy rate using the Kaggle Telecom Churn dataset, demonstrating its effectiveness in predicting customer retention outcomes [10].

C. He et al. (2024) proposed an Ensemble-Fusion model that incorporated 17 machine learning algorithms across nine categories to enhance churn prediction [11]. Using the Company's customer production line system dataset from 2015 to 2022, the model achieved 95.35% accuracy and an AUC score of 91%. The study demonstrated that ensemble-based approaches significantly improved prediction accuracy compared to single-model methods.

V. H. Vu et al. (2024) developed a stacked model for early churn detection in the banking industry [12]. This model was structured across two levels: the first level combined K-nearest neighbors, XGBoost, Random Forest, and Support Vector Machine, while the second level utilized logistic regression, recurrent neural networks, and deep learning networks to refine predictions. The approach achieved a high accuracy of 95.13% on the Kaggle Bank Turnover Dataset, outperforming traditional models in predictive performance and computational efficiency [12].

T. R. Noviandy et al. (2024) explored the application of LightGBM for churn prediction in the telecommunications sector [13]. The model achieved an accuracy of 80.70%, precision of 84.35%, and recall of 90.54% using the Kaggle Telco customer churn dataset. The study incorporated SHAP for model interpretability, identifying key churn factors such as contract type and monthly charges, thereby providing actionable insights for retention strategies.

A. Li et al. (2024) investigated the role of ensemble learning techniques in predicting customer churn in the telecommunications industry [14]. The study compared various machine learning models, emphasizing the effectiveness of the Stacking ensemble method. Results indicated that CatBoost achieved the highest accuracy at 81.19%, outperforming Random Forest (79.02%) and XGBoost (78.20%). The study highlighted CatBoost's superior performance due to its ability to handle categorical features effectively.

Despite the progress made by recent machine learning models in predicting customer churn, most existing solutions are limited by their reliance on batch-based training, which hinders adaptability to evolving data. Furthermore, current models rarely combine event-driven architectures with continual learning mechanisms, leaving a critical gap in real-time adaptability and long-term retention. This gap highlights the need for an integrated approach that supports dynamic learning from streaming data while maintaining model stability and accuracy. The proposed RCE model addresses this need by combining event-driven development, replay-based continual learning, and stacked ensemble classification.

III. Background and Key Concepts

The RCE model integrates event-driven development, which enables real-time data processing, replay-based continual learning, which adapts to evolving tasks while mitigating catastrophic forgetting, and stacked ensemble learning, a powerful classification technique for customer churn prediction. The following sub-sections provide an in-depth explanation of these key components, highlighting their significance within the proposed model to establish a solid foundation for the research.

A. *Event-Driven Development*

Event-Driven Development (EDD) is a software architectural paradigm that enables systems to respond to events in real-time. It is widely used in real-time data processing, IoT applications, and customer churn analysis, where continuous monitoring and swift decision-making are required [15].

The key components required to build a powerful event-driven architecture include:

1) Producers (Event sources). Entities that generate events based on real-world activities or system changes. For instance, in a customer churn analysis system, a producer could be a transaction monitoring system that detects abnormal customer activity [15, 16].

2) Event brokers (Middleware). A messaging infrastructure that ensures events are transmitted reliably between producers and consumers. Message brokers like Apache Kafka, RabbitMQ, or AWS Kinesis act as intermediaries, enabling scalable and decoupled communication [15, 16].

3) Consumers (Event processors). Systems or services that subscribe to and process events, executing appropriate responses. In churn prediction, a consumer could be an AI-driven analytics engine that processes incoming customer activity data and triggers predictive churn alerts [15, 16].

4) Event store (Optional). A repository that records past events for auditing, debugging, or replaying historical event sequences to improve machine learning model performance [16].

For instance, a simplified event-driven architecture is illustrated in Fig. 2, where multiple producers send real-time event data to a broker, which then distributes them to different consumers for processing.

B. *Replay-Based Continual Learning*

Continual Learning (CL) enables machine learning models to learn from a continuous stream of data while retaining knowledge from previous tasks. One of the most effective techniques for mitigating catastrophic forgetting in CL is the Replay-Based Approach, which involves storing past experiences and revisiting them periodically during training [17]. There are two primary types of replay-based continual learning:

1) Experience replay: This technique stores and reuses previous training examples to reinforce learning [17, 18].

a) Uniform replay: Randomly selects past samples for retraining [17].

b) Selective replay: Stores only important samples based on criteria such as uncertainty or importance weighting [17].

c) Prioritized replay: Assigns priority scores to samples based on their learning significance, ensuring high-value experiences are replayed more often [17].

2) Generative replay: Instead of storing actual data, generative models such as Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs) generate synthetic samples resembling past data distributions, allowing the model to reconstruct and recall previous knowledge efficiently [17].

Replay-based is effective in continual learning, where past experiences are retrieved and mixed with new data to maintain performance stability over time.

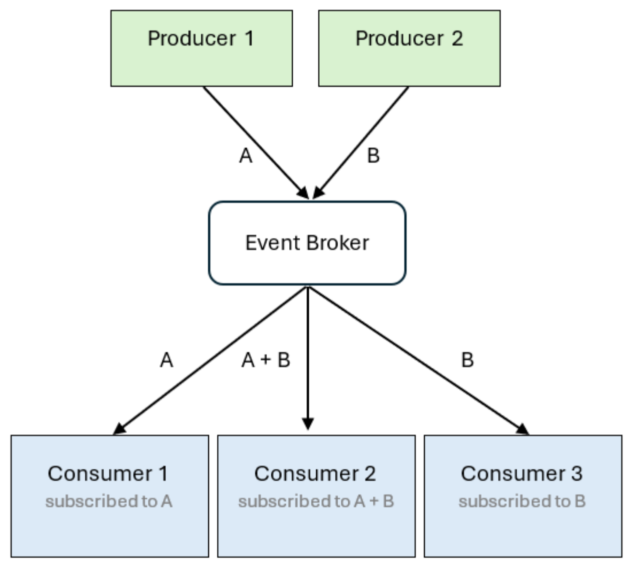

Fig. 2. A Simplified event-driven architecture.

C. *Stacked Ensemble Learning*

Stacked Ensemble Learning is an advanced machine learning technique that enhances predictive performance by combining multiple models, known as base learners, and a secondary model, called a meta-learner, to optimize decision-making as shown in Fig. 3. This approach leverages the strengths of diverse base models while mitigating their individual weaknesses, resulting in improved accuracy and robustness [19].

The key components required to build a powerful stacked ensemble learning model include:

1) Base models. Also referred to as level 1 learners, are diverse machine learning algorithms trained independently on the same dataset. Each base model captures unique patterns and relationships within the data, making different errors and offering complementary perspectives [19]. Common base models used in stacked ensemble learning include:

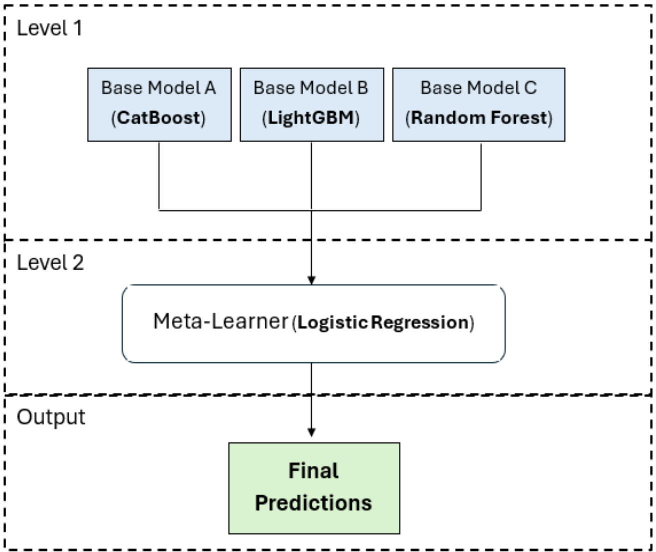

Fig. 3. A Stacked Ensemble Learning framework.

a) CatBoost: A gradient boosting algorithm optimized for categorical data [21].

b) XGBoost: An efficient and scalable implementation of gradient boosting [20].

c) LightGBM: A high-speed gradient boosting framework with optimized memory usage [22].

d) Random forest: An ensemble of decision trees that enhances generalization [19].

The diversity among base models ensures that they learn varying feature interactions and decision boundaries, reducing the risk of overfitting to specific data distributions.

2) Meta-learner. Also referred to as level 2 model, is responsible for aggregating and refining predictions from the base models. Instead of making direct predictions on the raw dataset, it takes the outputs (predicted probabilities or class labels) of the base models as input features and learns how to optimally combine them [19]. The meta-learner typically uses:

a) Logistic regression: A simple yet effective model for binary classification tasks [23].

b) Neural networks: When nonlinear interactions between base models need to be captured.

c) Gradient boosting models: If additional feature transformations are necessary [20, 21].

By learning the strengths and weaknesses of each base model, the meta-learner assigns appropriate weights to their predictions, ultimately improving classification accuracy.

IV. The Proposed Model (RCE)

Real-time Continual Ensemble (RCE) model is an advanced machine learning proposed model designed to enhance customer churn prediction through a combination of event-driven development, replay-based continual learning, and stacked ensemble learning, as shown in Fig. 4.

The proposed RCE model continuously processes incoming customer interaction data, dynamically adapts to changing behavioral patterns, and refines its predictive performance over time. The event-driven architecture is implemented using Apache Kafka, ensuring seamless real-time data ingestion from multiple sources, such as customer transactions, service usage logs, and customer support interactions [15, 16]. Each event triggers an update cycle in the RCE model, allowing for continuous learning.

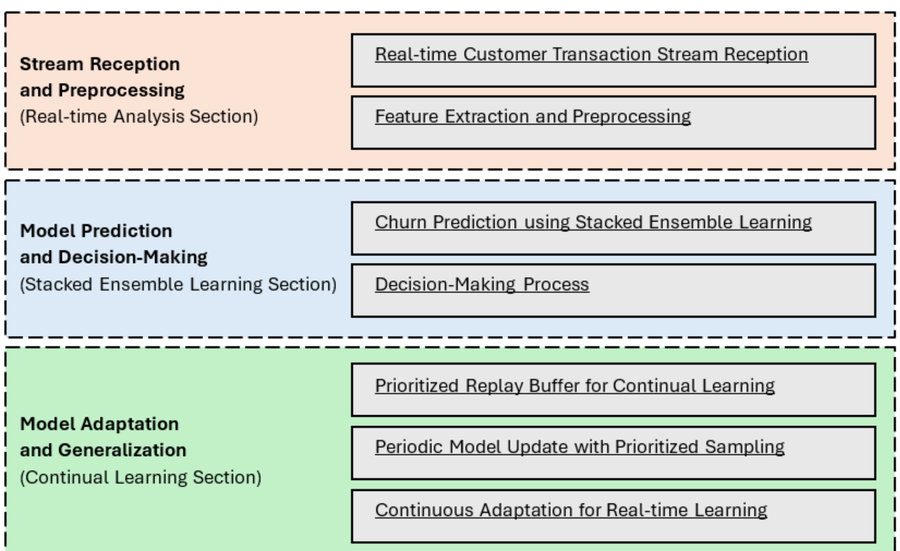

Fig. 4. The Proposed RCE model

To handle catastrophic forgetting and ensure knowledge retention, RCE integrate a prioritized experience replay mechanism, where past data points are stored in a buffer with priority scores based on their significance [17, 18]. When retraining, the model selectively samples from this buffer, focusing on high-impact instances to reinforce learning. At the core of the RCE model, lies a stacked ensemble learning approach, which combines multiple base classifiers—CatBoost, LightGBM, and Random Forest—to generate diverse predictions. A Logistic Regression model serves as the meta-learner, aggregating the base models' outputs to enhance predictive accuracy as shown in Fig. 3. The model development process includes comprehensive data preprocessing (handling missing values, feature extraction, normalization), feature engineering (polynomial transformations, encoding categorical variables), addressing class imbalance using ADASYN [24], and feature selection using a Random Forest-based method. To optimize performance, Bayesian Optimization is applied for hyperparameter tuning [25], systematically searching for the most effective parameter configurations for each base model as shown in Algorithm 1.

Algorithm 1: RCE Proposed Model

Input: Real-time customer interactions, such as service usage, transaction records, customer support interactions, or website activity. Events are generated by producers (e.g., transaction systems, CRM systems, or log monitoring services).

Output: Real-time churn probability scores, for each customer event, the model outputs a probability indicating the likelihood of churn. For instance, 80% (customer will stay) and 20% (customer will leave). To convert this probability into a binary decision, a threshold (e.g., 0.5 or 0.7) is typically applied (e.g., if the score $\geqslant$ 0.5, classified as 1 (churn) and the system can trigger proactive interventions, otherwise classify as 0 (stay)). The threshold can be adjusted based on business needs to optimize precision and recall.

Algorithm:

```
Initialize Event-driven data pipeline
Initialize Prioritized Replay Buffer (B) with Capacity (C)
Initialize Base models: CatBoost, LightGBM, and Random Forest with predefined hyperparameters
Initialize Meta-learner: Logistic Regression
While system is operational:
    | Receive new customer event (Et) from Kafka
    | Extract and preprocess features from Et:
    |       | Handle missing values, noises and outliers
    |       | Perform scaling, encoding, and feature transformation
    |       | Apply polynomial feature expansion
    |       | Apply feature selection techniques
    | Store processed event data in buffer B with priority score
    | If buffer B reaches capacity C:
    |       | Remove lowest-priority instances
    | If training condition met:
    |       | Sample mini-batch S from B by priority-based sampling
    |       | Apply (ADASYN) to balance class distribution within S
    |       | Tune hyperparameters using Bayesian Optimization
    |       | Train base models on S
    |       | Generate meta-features from base model outputs
    |       | Train meta-learner using meta-features
    | Predict customer churn probability score P(Et)
    | If P(Et) > threshold:
    |       | Trigger customer retention strategy
    | Update priorities in B based on instance importance
Return: Continuously updated RCE model
```

The selection of CatBoost, LightGBM, and Random Forest as base models, along with Logistic Regression as the meta-learner, is driven by their complementary strengths in customer churn prediction. CatBoost and LightGBM are boosting-type algorithms, meaning they build models sequentially, where each iteration corrects the errors of the previous one, leading to a strong overall model. Boosting is particularly effective in reducing bias and improving accuracy. Random Forest, on the other hand, is a bagging-type algorithm, which constructs multiple decision trees independently in parallel and then averages their predictions, enhancing stability and reducing variance. CatBoost efficiently handles categorical variables without extensive preprocessing, making it ideal for customer-related data [21]. LightGBM is optimized for speed and scalability, making it well-suited for large datasets [22]. Random Forest provides robustness against overfitting by leveraging an ensemble of decision trees [19]. The diversity among these models ensures that different aspects of the data are captured, reducing model bias and improving generalization. Logistic Regression, serving as the meta-learner, aggregates predictions from the base models, refining the final decision boundary [23]. This stacked ensemble approach enhances predictive performance, leading to a more accurate and reliable churn prediction system.

The entire process operates in a continuous learning loop, ensuring that the RCE model evolves with changing customer behaviors. As new data arrives, the model updates its knowledge while maintaining past learnings, effectively mitigating the challenges of traditional batch learning approaches. The proposed RCE model bridges real-time processing, continual learning, and stacked ensemble techniques, ensuring superior adaptability and predictive accuracy.

By integrating Kafka-based event-driven architectures, replay-based learning, and a highly optimized ensemble framework, the RCE model not only enhances churn prediction performance but also maintains robustness in dynamic customer environments.

V. Experimental Results

To assess the effectiveness of the RCE model in predicting customer churn, a series of experiments has been conducted using the "Churn Modelling" benchmark dataset for European banks, available on Kaggle [6]. This dataset comprises 10,000 customer records and includes 14 features, such as customer demographics, credit scores, number of products, estimated salary, and account activity, which are essential in predicting customer churn. The target variable, "Exited", indicates whether a customer has churned (1) or remained (0). Given the class imbalance in the dataset, where most customers do not churn, the dataset required specific preprocessing techniques to ensure balanced learning.

A. Experimental Setup

The RCE model is implemented in a controlled experimental environment with Kafka (v3.9) facilitating real-time event-driven processing. The machine learning components were developed using Python with the following libraries and frameworks: CatBoost (v1.2), LightGBM (v4.1.0), Scikit-learn (v1.3.0) for Random Forest and Logistic Regression, and Imbalanced-learn (v0.11) for applying ADASYN to address class imbalance. Bayesian Optimization was performed using Scikit-Optimize (v0.9.0) to fine-tune hyperparameters. The StackingClassifier from Scikit-learn was utilized to integrate the base learners (CatBoost, LightGBM, and Random Forest) with a Logistic Regression meta-learner. Data preprocessing involved feature engineering, feature selection using SelectFromModel, and polynomial feature augmentation to enhance model representation. The computational experiments were conducted on a system equipped with an Intel Core i7 processor, 32GB RAM, and an NVIDIA RTX 3090 GPU, ensuring efficient model training and inference. The performance of the RCE model is assessed under two experimental scenarios to show the effectiveness of the proposed models in different environments.

B. Scenario 1: Stable Environment

In the first scenario, the entire dataset was used as a single batch for learning and evaluation. The model is trained on a training subset (typically 70 to 80% of the dataset) and then evaluated on the remaining test data in a static, traditional batch setting. The RCE model demonstrated superior performance, achieving an accuracy of 95.65%, a precision of 90.41%, a recall (sensitivity) of 88.29%, and an F1-score of 89.71% as shown in Fig. 5.

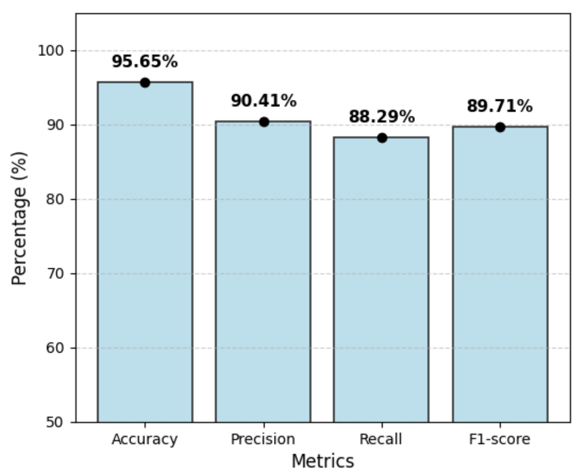

Fig. 5. RCE Performance metrics.

For comparative analysis, the RCE model's accuracy was evaluated against two recent customer churn prediction models that used the same dataset. The XGBoost model [8] achieved an accuracy of 83%, while the stacked ensemble model [12] attained 95.13% as shown in Fig. 6.

C. Scenario 2: Dynamic Environment

In the second scenario, the accuracy of the RCE model was evaluated in a simulated dynamic environment. Instead of using the entire dataset as a single batch in learning and evaluation, the model is first trained on an initial 70% training set, while the remaining 30% is used to simulate a real-time, dynamic environment by being gradually introduced in smaller batches incrementally to simulate real-time learning. The accuracy of the RCE model across four tasks (T1, T2, T3, and T4) was recorded as 89%, 86%, 85%, and 87%, respectively. In contrast, traditional models fail to adapt and generalize to new experiences, as they lack the ability to incrementally train with newly arriving data.

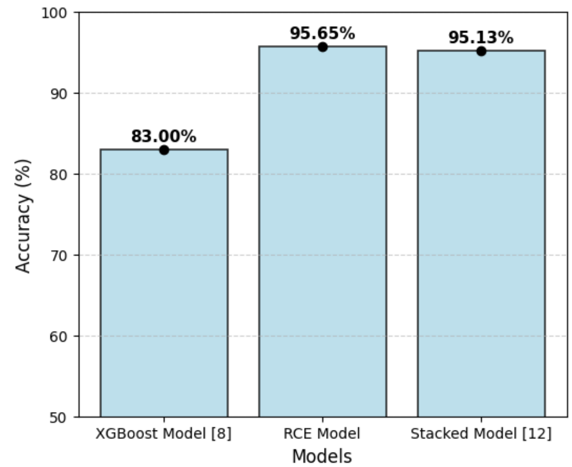

Fig. 6. Comparison of RCE with XGBoost [8] and Stacked Model [12] under static conditions.

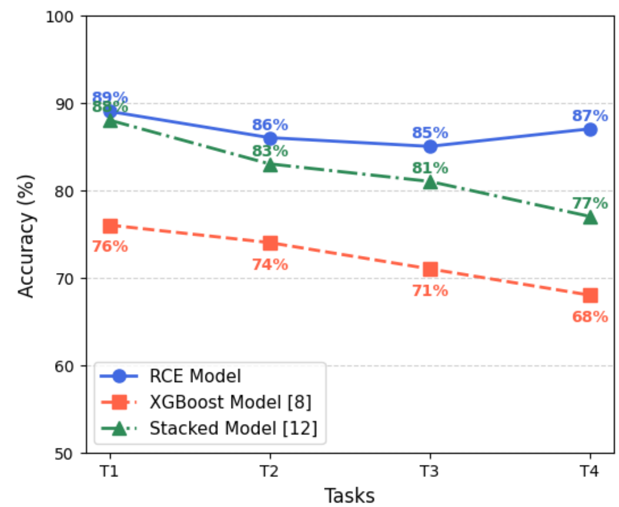

Fig. 7. Performance of RCE, XGBoost [8], and Stacked Model [12] across multiple tasks in a simulated dynamic environment.

Consequently, their performance deteriorates over time. The XGBoost model [8] exhibited a sharp decline in accuracy, scoring 76%, 74%, 71%, and 68%, while the stacked model [12] achieved 88%, 83%, 81%, and 77%. These results, illustrated in Fig. 7, demonstrate the effectiveness of the RCE model in maintaining robust predictive performance in dynamic environments.

To summarize the overall performance across dynamic tasks, the average accuracy (ACC) is computed. The RCE model achieved an average accuracy of 86.75%, whereas XGBoost [8] exhibited an average accuracy of 72.25%. The stacked model [12] had an average accuracy of 82.25% as shown in Fig. 8. The experimental results demonstrate that the RCE model significantly outperforms existing customer churn prediction models, both in traditional batch learning and dynamic real-time environments. By integrating event-driven processing, replay-based continual learning, and stacked ensemble learning, the RCE model maintains high predictive accuracy and effectively adapts to evolving customer behaviors, mitigating catastrophic forgetting.

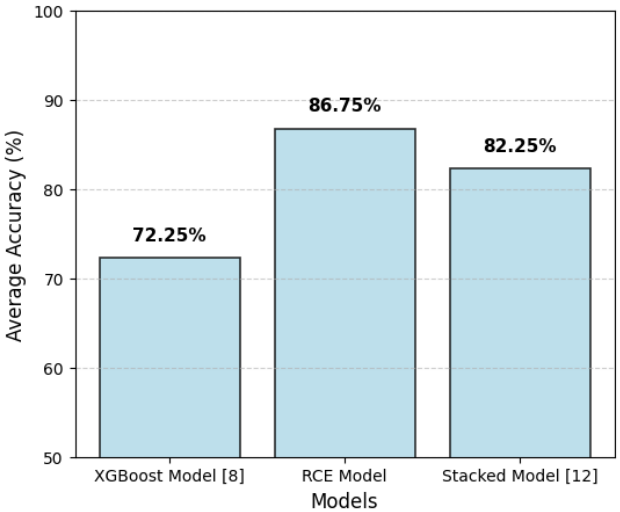

Fig. 8. Average accuracy of RCE, XGBoost [8], and Stacked Model [12] across multiple tasks in a simulated dynamic environment.

VI. Conclusion, Limitations, and Future Work

This research introduced the Real-time Continual Ensemble (RCE) model, designed to address customer churn prediction in real-time, dynamic environments. The RCE model integrates an event-driven development approach using Kafka to process continuous data streams efficiently, ensuring real-time churn prediction. Additionally, it incorporates a replay-based continual learning mechanism, allowing the model to adapt incrementally to evolving customer behaviors while mitigating catastrophic forgetting. Unlike traditional models that train on static datasets, the RCE model continuously refines its knowledge without losing prior insights. Furthermore, stacked ensemble learning was employed to enhance classification performance, leveraging a diverse set of base models (CatBoost, LightGBM, and Random Forest) and a Logistic Regression meta-learner. This combination reduces bias, enhances generalization, and improves predictive accuracy.

To evaluate the effectiveness of the proposed model, experiments were conducted on the Churn Modelling benchmark dataset for European banks, which consists of 10,000 samples and 14 features. The RCE model achieved a 95.65% accuracy when trained on the entire dataset, outperforming existing models. However, in a simulated dynamic environment, where incremental learning was required, RCE demonstrated superior adaptability by maintaining an average accuracy (ACC) of 86.75% across tasks. This performance was notably higher compared to recent models, such as the XGBoost model [8] which struggled with adapting to evolving data distributions, achieving only 72.25% ACC. Similarly, the stacked ensemble model [12], though initially competitive, exhibited a decline over time, reaching an ACC of 82.25%. The findings confirm that RCE not only outperforms traditional models in static training scenarios but also excels in dynamic environments, ensuring that businesses can make data-driven decisions in real time.

Despite its advantages, the proposed RCE model has certain limitations. First, the dataset used for evaluation is relatively small (10,000 samples), which may not fully capture the complexities of real-world customer churn behavior in large-scale enterprises. Additionally, the experience replay mechanism used for continual learning requires significant memory resources, especially when handling large volumes of streaming data. While prioritized experience replay helps mitigate this issue, memory constraints remain a challenge in real-time applications. Moreover, hyperparameter tuning using Bayesian Optimization contributes to enhanced model performance but is computationally expensive, requiring careful optimization to balance efficiency and accuracy.

Future research should focus on evaluating RCE on larger, more diverse datasets to assess its scalability and robustness in real-world scenarios. Furthermore, while experience replay has been effective in this study, alternative continual learning strategies could be explored to further enhance adaptability. Approaches such as gradient-based methods [26], regularization techniques [27], knowledge distillation [28], Bayesian-based adaptation [29], architecture modifications [30], or hybrid continual learning models offer promising directions to reduce memory overhead and improve long-term retention of knowledge. By addressing these challenges, the RCE model can be extended into broader applications, ensuring its effectiveness in high-velocity, real-time data-driven decision-making systems.

References

[1] A. Manzoor, M. Atif Qureshi, E. Kidney and L. Longo, "A Review on Machine Learning Methods for Customer Churn Prediction and Recommendations for Business Practitioners," in IEEE Access, vol. 12, pp. 70434-70463, 2024, doi: 10.1109/ACCESS.2024.3402092.

[2] Ransom, C. (2021). The cost of customer acquisition versus retention in fintech. Harvard Business Review. Retrieved from https://hbr.org/2021/03/the-cost-of-customer-acquisition-versus-retention-in-fintech.

[3] Shobana, J., Ch Gangadhar, Rakesh Kumar Arora, P. N. Renjith, J. Bamini, and Yugendra devidas Chincholkar. "E-commerce customer churn prevention using machine learning-based business intelligence strategy." Measurement: Sensors 27 (2023): 100728.

[4] Adeniran, Ibrahim Adedeji, Christianah Pelumi Efunniyi, Olajide Soji Osundare, Angela Omozele Abhulimen, and U. OneAdvanced. "Implementing machine learning techniques for customer retention and churn prediction in telecommunications." Computer Science & IT Research Journal 5, no. 8 (2024).

[5] Amin, Adnan, Awais Adnan, and Sajid Anwar. "An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and Naïve Bayes." Applied Soft Computing 137 (2023): 110103.

[6] S. Iyyer, “Churn Modelling,” Kaggle, 2019. [Online]. Available: https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling

[7] Chang, Victor, Karl Hall, Qianwen Ariel Xu, Folakemi Ololade Amao, Meghana Ashok Ganatra, and Vladlena Benson. 2024. "Prediction of Customer Churn Behavior in the Telecommunication Industry Using Machine Learning Models" Algorithms 17, no. 6: 231. doi: 10.3390/a17060231.

[8] Singh, Pahul Preet, Fahim Islam Anik, Rahul Senapati, Arnav Sinha, Nazmus Sakib, and Eklas Hossain. "Investigating customer churn in banking: A machine learning approach and visualization app for data

science and management." Data Science and Management 7, no. 1 (2024): 7-16. doi: 10.1016/j.dsm.2023.09.002

[9] Poudel, Sumana Sharma, Suresh Pokharel, and Mohan Timilsina. "Explaining customer churn prediction in telecom industry using tabular machine learning models." Machine Learning with Applications 17 (2024): 100567. doi: 10.1016/j.mlwa.2024.100567

[10] Akhmetshin, Elvir, Nurulla Fayzullaev, Elena Klochko, Denis Shakhov, and Valentina Lobanova. "Intelligent Data Analytics using Hybrid Gradient Optimization Algorithm with Machine Learning Model for Customer Churn Prediction." Fusion: Practice & Applications 14, no. 2 (2024).

[11] He, Chenggang, and Chris HQ Ding. "A novel classification algorithm for customer churn prediction based on hybrid Ensemble-Fusion model." Scientific Reports 14, no. 1 (2024): 20179. doi: 10.1038/s41598-024-71168-x

[12] Vu, Van-Hieu. "An efficient customer churn prediction technique using combined machine learning in commercial banks." In Operations research forum, vol. 5, no. 3, p. 66. Cham: Springer International Publishing, 2024.

[13] Noviandy, Teuku Rizky, Ghalieb Mutig Idroes, Irsan Hardi, Mohd Afjal, and Samrat Ray. "A Model-Agnostic Interpretability Approach to Predicting Customer Churn in the Telecommunications Industry." Infolitika Journal of Data Science 2, no. 1 (2024): 34-44. doi: 10.60084/ijds.v2i1.199

[14] Li, Ang, Tianyi Yang, Xiaoan Zhan, Yadong Shi, and Huixiang Li. "Utilizing Data Science and AI for Customer Churn Prediction in Marketing." Journal of Theory and Practice of Engineering Science 4, no. 05 (2024): 72-79. doi: 10.53469/jtpes.2024.04(05).10

[15] Badgujar, Pooja. "Implementing Event-Driven Architectures for Real-Time Insights." Journal of Technological Innovations 5, no. 1 (2024). doi: 10.93153/wmtbmr58.

[16] Khriji, Sabrine, Yahia Benbelgacem, Rym Chéour, Dhouha El Houssaini, and Olfa Kanoun. "Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks." The Journal of Supercomputing 78, no. 3 (2022): 3374-3401.

[17] Rolnick, David, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. "Experience replay for continual learning." Advances in neural information processing systems 32 (2019).

[18] P. Buzzega, M. Boschini, A. Porrello and S. Calderara, "Rethinking Experience Replay: a Bag of Tricks for Continual Learning," 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, pp. 2180-2187, doi:10.1109/ICPR48806.2021.9412614.

[19] Zhou, Zhi-Hua, and Zhi-Hua Zhou. Ensemble learning. Springer Singapore, 2021. doi: 10.1007/978-981-15-1967-3_8

[20] Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, and Tianyi Zhou. "Xgboost: extreme gradient boosting." R package version 0.4-2 1, no. 4 (2015): 1-4.

[21] Prokhorenkova, Liudmila, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. "CatBoost: unbiased boosting with categorical features." Advances in neural information processing systems 31 (2018).

[22] Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).

[23] van Loon, Wouter, Marjolein Fokkema, Botond Szabo, and Mark de Rooij. "View selection in multi-view stacking: choosing the meta-learner." Advances in Data Analysis and Classification (2024): 1-39.

[24] Imani, Mehdi, Ali Beikmohammadi, and Hamid Reza Arabnia. 2025. "Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels" Technologies 13, no. 3: 88. doi:10.3390/technologies13030088

[25] Albahli, Saleh. "Efficient hyperparameter tuning for predicting student performance with Bayesian optimization." Multimedia tools and applications 83, no. 17 (2024): 52711-52735. doi: 10.1007/s11042-023-17525-w

[26] Lopez-Paz, David, and Marc'Aurelio Ranzato. "Gradient episodic memory for continual learning." Advances in neural information processing systems 30 (2017).

[27] Zhao, Xuyang, Huiyuan Wang, Weiran Huang, and Wei Lin. "A statistical theory of regularization-based continual learning." arXiv preprint arXiv:2406.06213 (2024).

[28] S. Li, T. Su, X. -Y. Zhang and Z. Wang, "Continual Learning With Knowledge Distillation: A Survey," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2024.3476068.

[29] Lee, Soochan, Hyeonseong Jeon, Jaehyeon Son, and Gunhee Kim. "Learning to continually learn with the Bayesian principle." arXiv preprint arXiv:2405.18758 (2024). doi:10.48550/arXiv.2405.18758

[30] Rusu, Andrei A., Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. "Progressive neural networks." arXiv preprint arXiv:1606.04671 (2016). doi:10.48550/arXiv.1606.04671