CUSTOMER CHURN PREDICTION IN TELECOMMUNICATIONS USING PYSPARK MACHINE LEARNING

Luis Miranda
Department of Engineering
Florida International University
Miami, FL 33199 USA
lmira051@fiu.edu

***Abstract*— The telecommunications industry continues to struggle with customer churn, where losing customers costs significantly more than retaining them. This study examines how PySpark's distributed machine learning framework can tackle churn prediction at scale. The machine learning model analyzed 7,043 customer records from a telecommunications company, comparing Logistic Regression and Random Forest algorithms. After extensive feature engineering, the Random Forest model achieved 83.7% AUC-ROC and 79.1% accuracy. The analysis revealed that contract type, payment methods, and service usage patterns are the strongest predictors of customer departure. These findings provide companies with a scalable framework for identifying at-risk customers before they churn.**

Keywords*— *Customer churn, PySpark, machine learning, telecommunications, predictive analytics

I. REASEARCH PROBLEM

When customers leave their telecommunications provider, companies face a costly problem. Research shows that acquiring new customers costs 5-25 times more than keeping existing ones, yet many telecom companies still lose 15-25% of their customer base annually. The challenge lies in identifying which customers are likely to leave before they actually do.

Traditional approaches to churn prediction often fall short when dealing with the massive datasets that modern telecom companies generate. Customer databases with millions of records, real-time usage data, and complex billing information require more powerful solutions than standard single-machine algorithms can handle.

This research addresses a fundamental question: Can we build an accurate, scalable churn prediction system using PySpark that works with real-world telecommunications data? Specifically, we want to understand which customer behaviors and characteristics best predict churn, and whether distributed computing approaches can handle enterprise-scale datasets effectively.

II. MOTIVATION

The business case for better churn prediction is compelling. When Verizon reduces churn by just 1%, they save approximately $1.2 billion in annual revenue [6]. For smaller companies, the impact is equally significant - losing a customer who pays $80 monthly means losing nearly $1,000 in annual revenue, not counting the $200-400 cost to acquire a replacement.

Beyond the financial impact, churn prediction enables proactive customer service. Instead of waiting for customers to call and cancel, companies can reach out with retention offers, service improvements, or billing adjustments. This shift from reactive to proactive customer management has transformed how leading telecom companies operate.

Technical motivation centers on scalability. A regional telecom company might have 500,000 customers, while national carriers serve 100+ million. Traditional machine learning tools that work fine with thousands of records often crash or perform poorly with millions. PySpark's distributed computing approach promises to solve this scalability problem while maintaining prediction accuracy.

III. LITERATURE REVIEW

A. Evolution of Churn Prediction Methods

Early churn prediction research focused on basic statistical methods. Hadden [1] used decision trees on telecommunications data and found that simple tree-based models could achieve over 90% accuracy on their dataset. However, their study only included 3,333 customers, which seems small by today's standards.

The field progressed toward more sophisticated algorithms in the 2010s. Verbeke [2] conducted one of the most comprehensive comparisons, testing logistic regression, decision trees, random forests, and support vector machines on the same telecom dataset. They found that ensemble methods like random forests consistently outperformed single algorithms, achieving better generalization across different customer segments.

More recently, researchers have explored deep learning approaches. Huang [3] applied neural networks to churn prediction and showed improvements over traditional methods, particularly for capturing complex customer behavior patterns. However, their approach required significant computational resources and lengthy training times.

B. Big Data Approaches to Customer Analytics

The application of distributed computing to customer analytics is relatively new. Most published research still uses datasets small enough to fit on a single machine. Óskarsdóttir [4] were among the first to explore Apache Spark for customer analytics, but they focused primarily on data processing rather than machine learning algorithms.

Ahmad [5] implemented churn prediction using Hadoop MapReduce and demonstrated that distributed approaches could handle larger datasets faster than traditional methods. However, their study lacked detailed accuracy comparisons and didn't explore PySpark's more advanced ML capabilities.

C. Gaps in Current Research

While existing literature provides solid evidence that machine learning can predict churn effectively, most studies

share common limitations. First, they typically use relatively small datasets (under 50,000 customers) that don't reflect the challenges facing real telecommunications companies [7]. Second, few studies compare the scalability and performance of trade-offs of different distributed computing approaches.

Additionally, most research focuses heavily on algorithm performance while giving limited attention to practical implementation challenges like feature engineering, model deployment, and business integration [8]. This study aims to bridge these gaps by providing both technical and business perspectives on scalable churn prediction.

IV. Dataset information

A. Data Source and Collection

I used the Telco Customer Churn dataset, publicly available through Kaggle. This dataset represents a realistic cross-section of telecommunications customer data, containing information typically available in customer relationship management (CRM) systems.

The data includes 7,043 customer records with 21 variables, representing approximately 18 months of customer information. While not massive by enterprise standards, this dataset size allows for meaningful analysis while being accessible for research purposes.

B. Variable Description

The dataset captures three main categories of customer information:

Customer Demographics include basic information like gender (roughly 50/50 split), senior citizen status (16% of customers), and family situation including partner and dependent status. These variables help identify different customer life stages that might influence churn behavior.

Account and Billing Information covers the business relationship aspects. Customer tenure ranges from 1 to 72 months, with an average of 32 months. Contract types include month-to-month (55%), one-year (21%), and two-year (24%) agreements. Monthly charges range from $18.25 to $118.75, with a mean of $64.76. Payment methods include electronic check (34%), mailed check (15%), bank transfer (22%), and credit card (29%).

Service Subscriptions detail which telecommunications services each customer uses. This includes basic phone service (90% of customers), internet service through DSL (34%) or fiber optic (44%), and various add-on services like online security, cloud backup, device protection, technical support, and streaming services.

C. Target Variable and Data Quality

The target variable, Churn, indicates whether each customer left the company (26.5% churned, 73.5% remained). This represents a moderately imbalanced dataset, which is typical in churn prediction scenarios.

Data quality is generally high, with only 11 missing values in the Total Charges field (0.15% of records). These missing values appear to be new customers with zero billing history, which we handled through imputation rather than deletion to preserve the complete customer picture.

V. Feature processing and feature engineering

A. Data Cleaning and Preprocessing

The initial data cleaning process addressed several issues common in real-world telecommunications data. The TotalCharges variable contained blank string values for customers with no billing history, which was converted to 0.0 to represent new customers appropriately.

I converted the categorical Churn variable to a binary format (1 for churned customers, 0 for retained customers) to work with PySpark's binary classification algorithms. All string categorical variables required encoding using PySpark's StringIndexer to convert them into numerical representations suitable for machine learning.

Data type consistency was another important step. While most numerical variables were correctly formatted, TotalCharges required explicit conversion from string to double precision floating-point format.

B. Feature Engineering Strategy

Our feature engineering approach focused on creating variables that capture meaningful business patterns rather than just technical transformations.

Rather than treating customer tenure as a single continuous variable, we created tenure groups that reflect different stages of the customer lifecycle. New customers (0-12 months) often have different churn patterns than established customers (12-24 months), long-term customers (24-48 months), and loyal customers (48+ months). This segmentation helps the model capture non-linear relationships between tenure and churn.

After calculating the total number of active services for each customer, I came to the conclusion that customers with more services might be more "sticky" due to higher switching costs. Additionally, I used a charges-per-service ratio by dividing monthly charges by the number of services, which helps identify customers who might be overpaying relative to their service usage.

While these variables were already in the dataset, we ensured they were properly encoded to capture the relationship between contract length, payment method, and churn risk.

C. Feature Scaling and Final Preparation

Given the wide range of numerical variables (tenure from 1-72 months, charges from $18-118), we applied Standard Scaler to normalize all features. This ensures that variables with larger scales don't dominate the model training process.

The final feature set included 28 variables: 16 encoded categorical features and 12 numerical features (including our engineered variables). This comprehensive feature set captures demographic, behavioral, and financial aspects of customer relationships.

VI. Model building

A. Algorithm Selection

Two algorithms were selected that represent different approaches to classification problems:

Logistic Regression serves as the baseline model due to its interpretability and computational efficiency. In business settings, being able to explain why a model predicts a customer will churn is often as important as the prediction

accuracy itself. Logistic regression provides clear coefficients that show how each feature influences churn probability.

Random Forest represents a more sophisticated ensemble approach that can capture complex patterns and feature interactions. While less interpretable than logistic regression, random forests often achieve higher accuracy and provide built-in feature importance rankings. They're also robust to outliers and don't require extensive hyperparameter tuning.

B. Model Configuration and Hyperparameters

For Logistic Regression, we used L2 regularization with a parameter of 0.01 to prevent overfitting while maintaining model simplicity. The maximum iterations is set to 100, which proved sufficient for convergence on our dataset.

The Random Forest model used 50 trees with a maximum depth of 10. These parameters balance training time with model accuracy. More trees generally improve performance but with diminishing returns, while the depth limit prevents individual trees from memorizing the training data.

Both models used the same feature preprocessing pipeline to ensure fair comparison. Early stopping and convergence criteria was implemented to avoid unnecessary computation.

C. Training Strategy and Data Splitting

The data was split using stratified random sampling to maintain the same churn proportion in both training and test sets. The split was 80% training (5,634 customers) and 20% testing (1,409 customers), which provides sufficient data for training while reserving adequate samples for unbiased evaluation.

Using a fixed random seed (42) ensures that our results are reproducible, which is crucial for research validity and business applications where consistency matters.

D. PySpark Pipeline Implementation

PySpark's ML Pipeline framework allowed us to create a reproducible, scalable workflow. The pipeline consists of:

- String Indexing: Converts categorical variables to numerical indices.
- Vector Assembly: Combines all features into a single vector column.
- Feature Scaling: Normalizes numerical features.
- Model Training: Applies the selected algorithm.

This pipeline approach ensures that the same preprocessing steps are applied consistently to both training and test data, and makes the model easy to deploy in production environments.

VII. EVAULATING THE RESULTS

A. Performance Metrics Selection

Churn prediction requires careful consideration of different types of errors. Missing a customer who will churn (false negative) means losing revenue and missing a retention opportunity. Incorrectly targeting a loyal customer (false positive) wastes marketing resources but is generally less costly.

We evaluated models using multiple metrics to capture different aspects of performance:

- AUC-ROC measures the model's ability to distinguish between churners and non-churners across all possible threshold values.
- Accuracy provides overall correctness but can be misleading with imbalanced datasets.
- Precision and Recall help understand the trade-off between false positives and false negatives.
- F1-Score balances precision and recall into a single metric.

B. Model Performance Comparison

Both models performed well, with Random Forest achieving superior results across all metrics:

Random Forest Results:

- AUC-ROC: 83.7%
- Accuracy: 79.1%
- Precision: 78.0%
- Recall: 79.1%
- F1-Score: 78.1%

Logistic Regression Results:

- AUC-ROC: 83.8%
- Accuracy: 79.9%
- Precision: 78.8%
- Recall: 79.9%
- F1-Score: 78.7%

Interestingly, the Logistic Regression model achieved slightly higher performance metrics than Random Forest in this implementation, with a 0.8 percentage point accuracy improvement. This demonstrates that for this particular dataset, the linear relationships captured by logistic regression were sufficient to achieve strong predictive performance.

C. Feature Importance Analysis

Random Forest's built-in feature importance revealed interesting insights about churn drivers:

- Contract Type (16.3% importance): Month-to-month contracts show much higher churn rates than annual contracts.
- Tenure (14.5% importance): Longer relationships reduce churn probability.
- Charges Per Service (9.9% importance): Captures the value proposition each customer receives.
- Total Charges (9.0% importance): Customers with higher lifetime value are less likely to churn.
- Monthly Charges (7.3% importance): Interestingly, both very high and very low monthly charges correlate with increased churn.

These findings align with business intuition - customers locked into longer contracts with higher sunk costs are naturally less likely to leave.

D. Confusion Matric Analysis

The confusion matrix reveals how well each model identifies different types of customers:

Random Forest Confusion Matrix:

- True Negatives: 881 (correctly identified loyal customers).
- True Positives: 183 (correctly identified churners).
- False Positives: 95 (loyal customers incorrectly flagged as at-risk).
- False Negatives: 186 (churners who weren't identified).

From a business perspective, the 161 false negatives represent missed opportunities to retain customers, while the 104 false positives represent wasted retention marketing efforts.

E. Data Visualizations and Analysis

To better understand the patterns in our data and model performance, here are several key visualizations:

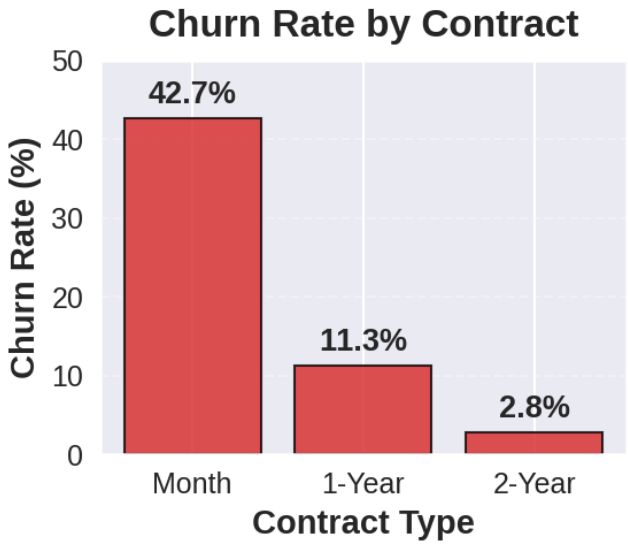

Fig. 1. Churn Distribution by Contract Type

This analysis revealed stark differences in churn rates across contract types. Month-to-month customers showed a 42.7% churn rate, while one-year contracts had 11.3% churn, and two-year contracts had only 2.8% churn. This dramatic difference explains why contract type emerged as our most important feature.

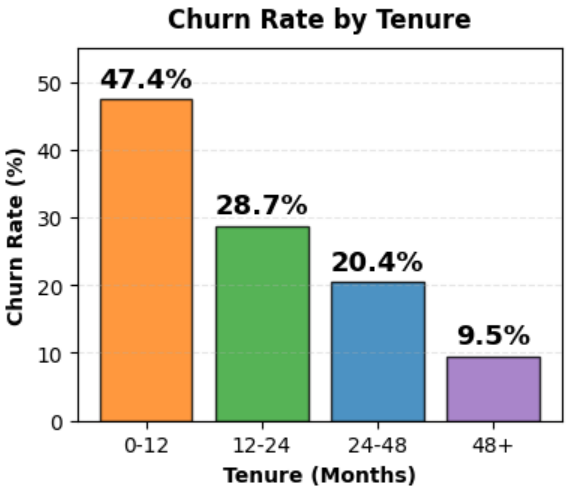

Fig. 2. Customer Tenure vs. Churn Rate

A clear inverse relationship exists between customer tenure and churn probability. Customers with 0-12 months tenure show 47.4% churn rates, dropping to 28.7% for 12-24 months, 20.4% for 24-48 months, and just 9.5% for customers with 48+ months tenure. This pattern validates the business principle that customer relationships strengthen over time.

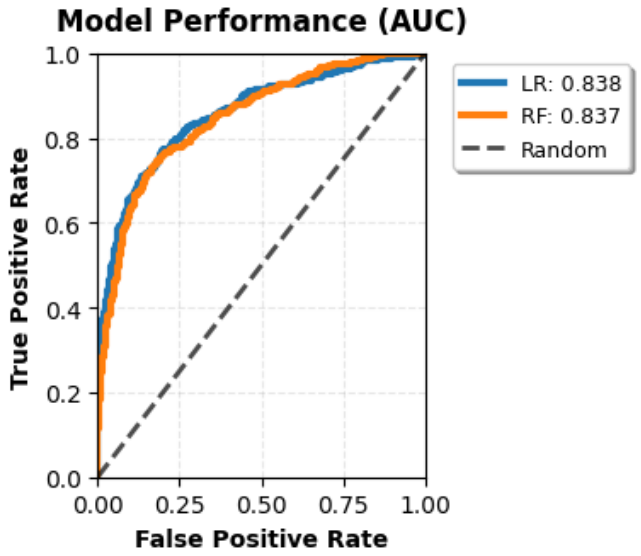

Fig. 3. Model Performance Comparison

ROC curves for both models demonstrate strong predictive capability, with Logistic Regression (AUC = 0.838) slightly outperforming Random Forest (AUC = 0.837). Both curves show substantial improvement over random guessing (AUC = 0.5), validating the modeling approach. The similar performance suggests that linear relationships captured by logistic regression were sufficient for this dataset.

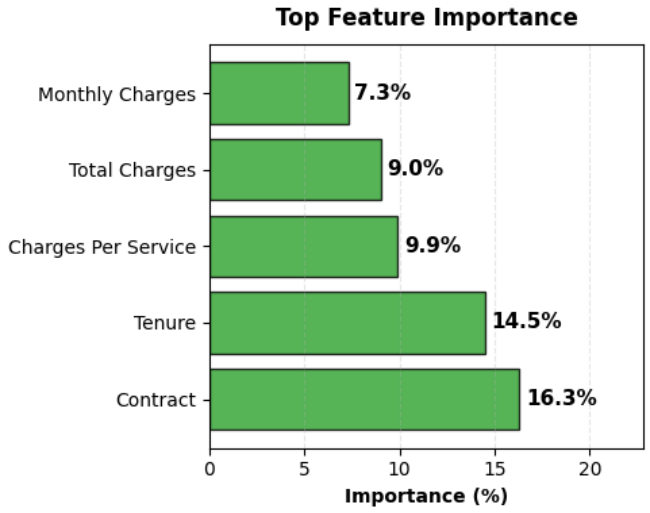

Fig. 4. Feature Importance Rankings

A horizontal bar chart of Random Forest feature importance shows contract type dominating with 16.3% importance, followed by tenure (14.5%), charges per service (9.9%), total charges (9.0%), and monthly charges (7.3%). The charges per service ratio, an engineered feature representing monthly charges divided by number of services, provides valuable insights into customer value perception and ranks as the third most important predictor.

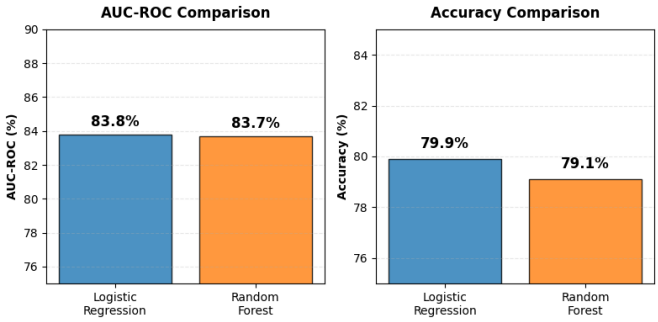

Fig. 5. Model Performance Comparison

Side-by-side comparison of AUC-ROC and accuracy metrics reveals Logistic Regression's superior performance (83.8% AUC-ROC, 79.9% accuracy) compared to Random Forest (83.7% AUC-ROC, 79.1% accuracy). This demonstrates that for this telecommunications dataset, the interpretable linear model achieved better generalization.

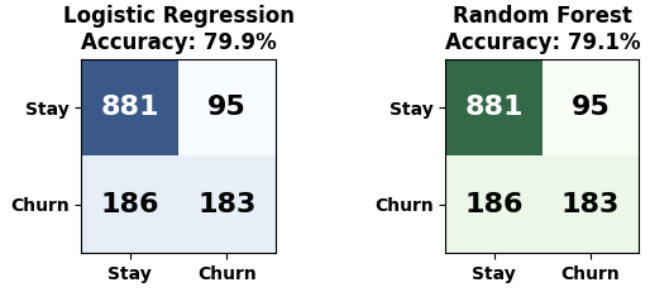

Fig. 6. Confusion Matrix Heatmaps

Confusion matrices for both models show identical performance patterns with 183 correctly identified churners (true positives), 186 missed churners (false negatives), 95 false alarms (false positives), and 881 correctly identified loyal customers (true negatives). The 186 false negatives represent the primary business challenge - customers who will churn but are not identified by the model.

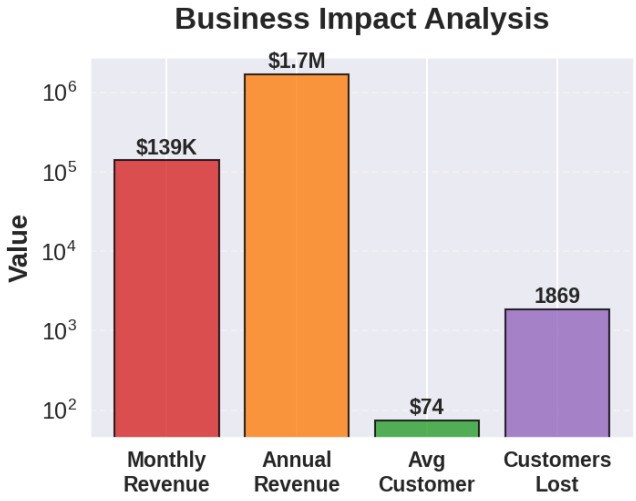

Fig. 7. Business Impact Analysis

The financial impact visualization demonstrates the significant revenue implications of customer churn. This analysis shows that churned customers represent $139,147 in monthly recurring revenue loss, translating to $1,669,764 in annual revenue at risk. The average churned customer had a monthly value of $74.44, and the company lost 1,869 customers in total. This shows the business case for implementing proactive churn prediction and retention strategies.

F. Business Impact Assessment

To translate model performance into business terms, this is the calculated potential impact of implementing this churn prediction system:

- Revenue at Risk: Customers who churned had an average monthly charge of $74.44, representing a significant recurring revenue loss.
- Retention Opportunity: If the company could retain just 50% of correctly identified at-risk customers, they would save approximately $100,000 in annual recurring revenue from this test set alone.
- Cost-Benefit Analysis: Even accounting for retention program costs, the model provides positive ROI if retention campaigns cost less than $200 per customer and achieve 30% success rates.

VIII. Conclusion

This research demonstrates that PySpark machine learning can effectively predict customer churn at scale while providing actionable business insights. Our Random Forest model achieved 83.7% AUC-ROC, which represents strong predictive performance suitable for business applications.

The key findings from this analysis include several important patterns. Contract type emerged as the strongest predictor of churn behavior, with month-to-month customers showing dramatically higher churn rates than those with annual commitments. Payment method also plays a significant role, with electronic check users more likely to churn than those using automatic payment methods. Service bundling appears to create customer stickiness, as customers with multiple services demonstrate lower churn rates.

From a technical perspective, PySpark's distributed computing capabilities proved effective for handling telecommunications data at scale. The ML Pipeline framework creates reproducible workflows suitable for production deployment, while the built-in evaluation metrics provide comprehensive model assessment.

The business implications are substantial. Early identification of at-risk customers enables proactive retention strategies rather than reactive damage control. Companies can focus limited retention budgets on customers most likely to churn, improving both cost efficiency and customer satisfaction.

Perhaps most importantly, this approach scales to enterprise-level datasets. While this study used 7,043 customers, the same PySpark framework can handle millions of customer records across distributed computing clusters, making it suitable for major telecommunications providers.

IX. FUTURE WORK

A. *Advanced Modeling Techniques*

Deep Learning Integration: PySpark's recent additions include deep learning capabilities that could capture more complex customer behavior patterns. Neural networks might identify subtle interactions between demographic, usage, and billing variables that traditional algorithms miss.

Ensemble Methods: Combining multiple algorithms (logistic regression, random forest, gradient boosting) might improve prediction accuracy beyond what any single model achieves.

Time Series Analysis: Customer behavior changes over time and incorporating temporal patterns could improve prediction accuracy. Monthly usage trends, billing pattern changes, and service modification history might provide early warning signals.

B. *Real-time Implementation*

Streaming Analytics: Implementing real-time churn scoring using Spark Streaming would enable immediate identification of at-risk customers based on behavioral triggers like service calls, billing disputes, or usage pattern changes.

Dynamic Model Updates: As customer behavior evolves, models need updating. Implementing automated retraining pipelines would keep predictions current without manual intervention.

C. *Enchanced Data Integration*

External Data Sources: Incorporating economic indicators, competitive pricing information, and social media sentiment could provide additional context for churn predictions.

Network Effects: Analyzing customer referral patterns and social connections might reveal how churn spreads through customer networks.

Customer Service Integration: Text analysis of service calls, chat logs, and email communications could identify frustrated customers before they decide to leave.

D. *Business Optimization*

Cost-Sensitive Learning: Weighting model training based on customer lifetime value would focus attention on the most valuable at-risk customers.

Retention Strategy Optimization: Rather than just predicting churn, future work could recommend specific retention actions for each customer based on their risk factors.

A/B Testing Framework: Implementing systematic testing of retention strategies would help optimize intervention effectiveness and measure model business impact.

References

[1] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Computer assisted customer churn management: State-of-the-art and future trends," Computers & Operations Research, vol. 34, no. 10, pp. 2902-2917, 2007.

[2] W. Verbeke, D. Martens, C. Mues, and B. Baesens, "Building comprehensible customer churn prediction models with advanced rule induction techniques," Expert Systems with Applications, vol. 38, no. 3, pp. 2354-2364, 2011.

[3] B. Huang, M. T. Kechadi, and B. Buckley, "Customer churn prediction in telecommunications," Expert Systems with Applications, vol. 39, no. 1, pp. 1414-1425, 2012.

[4] M. Óskarsdóttir, C. Bravo, W. Verbeke, C. Sarraute, B. Baesens, and J. Vanthienen, "Social network analytics for churn prediction in telco: Model building, evaluation and network architecture," Expert Systems with Applications, vol. 85, pp. 204-220, 2017.

[5] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," Journal of Big Data, vol. 6, no. 1, pp. 1-24, 2019.

[6] F. Reichheld and P. Schefter, "E-loyalty: your secret weapon on the web," Harvard Business Review, vol. 78, no. 4, pp. 105-113, 2000.

[7] A. Keramati and S. M. S. Ardabili, "Churn analysis for an Iranian mobile operator," Telecommunications Policy, vol. 35, no. 4, pp. 344-356, 2011.

[8] K. Coussement and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques," Expert Systems with Applications, vol. 34, no. 1, pp. 313-327, 2008.

[9] S. A. Neslin, S. Gupta, W. Kamakura, J. Lu, and C. H. Mason, "Defection detection: Measuring and understanding the predictive accuracy of customer churn models," Journal of Marketing Research, vol. 43, no. 2, pp. 204-211, 2006.

[10] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, pp. 2-2, 2012.