Real-Time Anomaly Detection and Forecasting in Streaming Platform Engagement using Time Series

By: Govardhan Reddy
and
Anand Kumar

INTRODUCTION

In today’s digital world, understanding user engagement is crucial for platforms like Netflix and YouTube. This project focuses on analyzing time series data to forecast user activity and detect anomalies that may indicate issues like user churn or abnormal behavior.

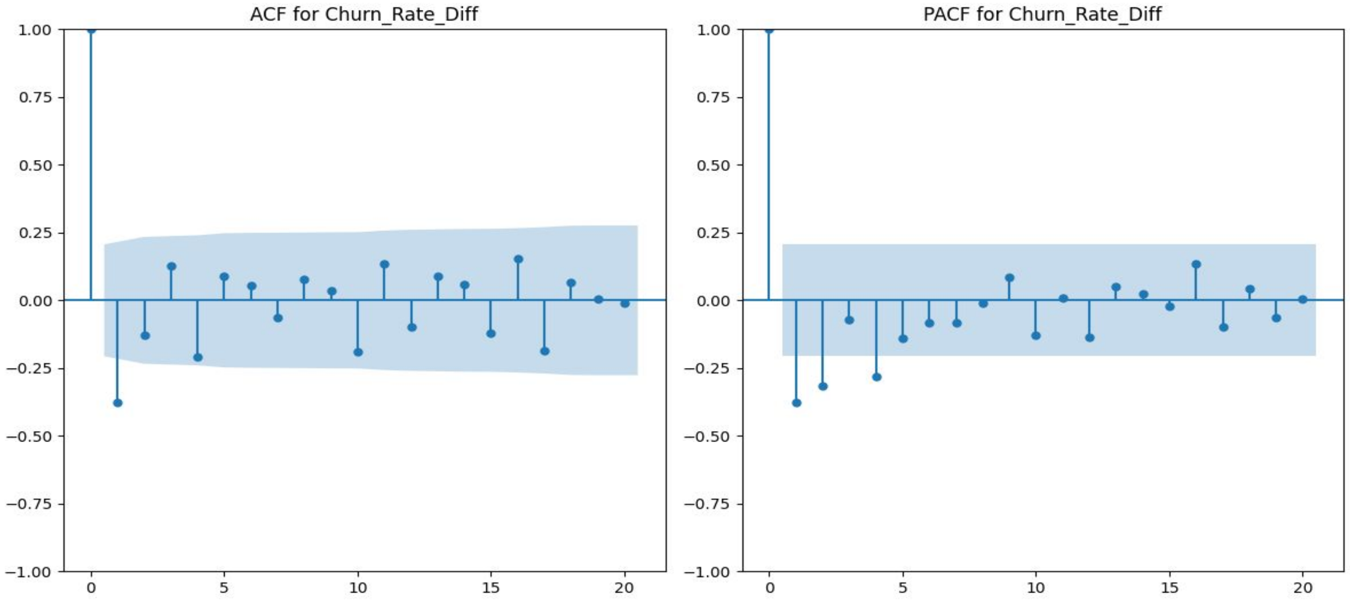
We used:

- ARIMA for statistical forecasting,
- LSTM for deep learning-based prediction
- Isolation Forest for identifying outliers in user behavior.

By combining these models, we can improve platform insights, support proactive decision-making, and enhance user retention strategies.

ARIMA

Plot for AutoCorrelation Function and Partial AutoCorrelation Function

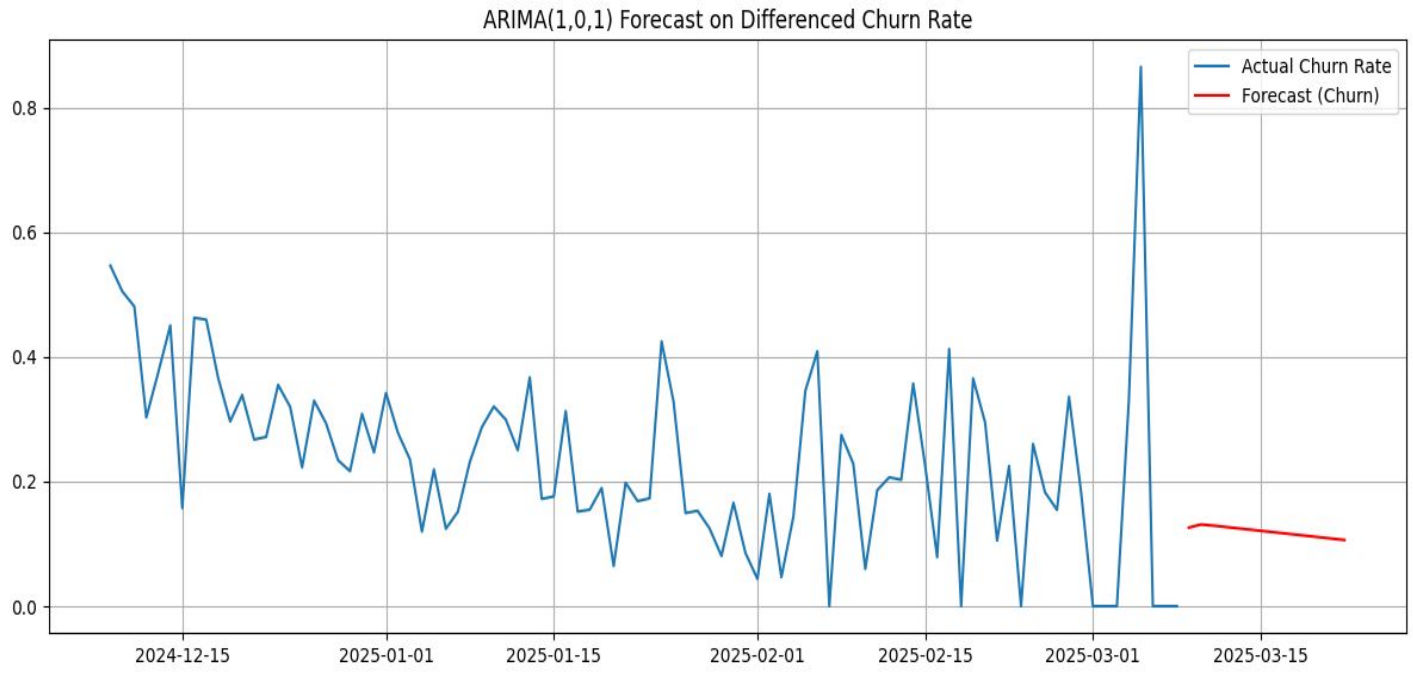


ARIMA INSIGHTS

Plot for AutoCorrelation Function and Partial AutoCorrelation Function

- Objective: To assess autocorrelation structure and identify appropriate ARIMA parameters (p, d, q).
.
- **ACF Insights:**
  - Rapid decay after lag 1.
  - Minor positive spikes around lags 2–10.
  - Suggests a short memory process → candidate MA(1) model.
- **PACF Insights:**
  - Significant drop after lag 1.
  - Clear cutoff → potential AR(1) model.

ARIMA PLOT



ARIMA(1,0,1) Forecast on Differenced Churn Rate
Actual Churn Rate
Forecast (Churn)
0.0
0.2
0.4
0.6
0.8
2024-12-15
2025-01-01
2025-01-15
2025-02-01
2025-02-15
2025-03-01
2025-03-15

ARIMA PLOT INSIGHTS

ARIMA(1,0,1) Forecast

- **Model Used**: ARIMA(1,0,1) on churn rates .
- **Training Period**: Historical Churn Rate till March 8, 2025.
- **Forecast Horizon**: 14-day forecast from March 9–22, 2025.
- **Forecast Pattern**:
  - Smooth and slightly declining trend.
  - Fails to capture recent spikes and drops (e.g., early March anomalies).
- **Limitation**:
  - ARIMA underfits non-linear fluctuations in Churn Rate.
  - Less responsive to sudden surges or drops (seen from flat red forecast).
- **Conclusion**:
  - While ARIMA provides stable forecasts, it lacks precision during volatile periods.

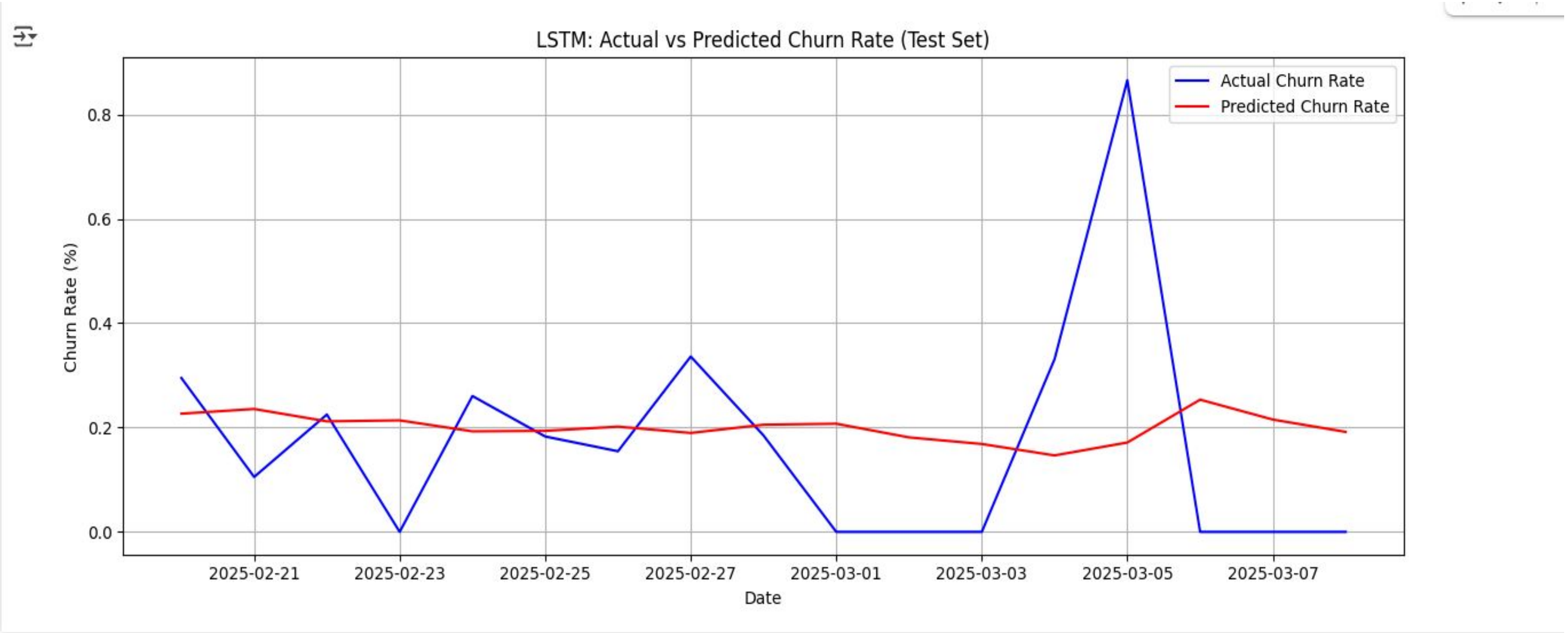
Interpretation:

- Ideally, residuals should appear as random noise (no clear pattern).
- This plot shows noticeable spikes, especially around early March.
- Suggests ARIMA(1,0,1) couldn't fully capture the volatility or anomalies in Churn Rate.

Implication:

- Residual spikes indicate underfitting or misspecification of the model.
- Performance might degrade during anomaly-heavy periods.

LSTM


LSTM: Actual vs Predicted Churn Rate (Test Set)
Actual Churn Rate
Predicted Churn Rate
Churn Rate (%)
0.0
0.2
0.4
0.6
0.8
2025-02-21
2025-02-23
2025-02-25
2025-02-27
2025-03-01
2025-03-03
2025-03-05
2025-03-07
Date

LSTM Churn Rate

- **Blue Line**: Actual churn rates from the test set.
- **Red Line**: Churn rates predicted by the trained LSTM model.
- **Performance Observation**:
  - LSTM **smoothly tracks** general churn behavior.
- **Strengths**:
  - Handles noisy, non-linear patterns better than ARIMA.
  - Maintains consistent performance despite fluctuations.

INSIGHTS

ARIMA:

- MAE (Mean Absolute Error): **0.1753**
- Struggles with sharp fluctuations and anomalies.
- Better suited for **linear, stationary trends**.

LSTM :

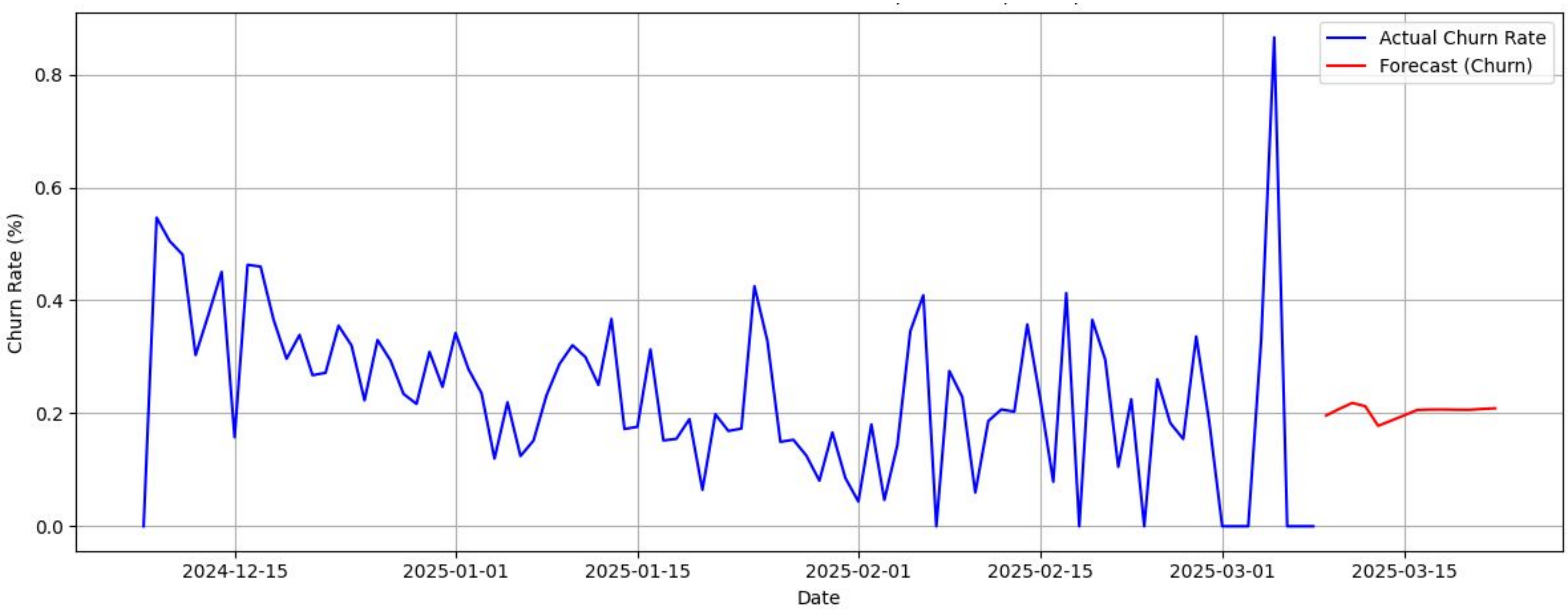
- MAE: **0.1651** (Lower than ARIMA)
- Better at capturing **non-linear patterns** and **temporal dependencies**.
- More resilient to **volatility and noise** in churn behavior.

```python
# Calculate metrics
rmse_total = np.sqrt(mean_squared_error(y_test_total_inv, y_pred_total_inv))
mae_total = mean_absolute_error(y_test_total_inv, y_pred_total_inv)
rmse_churn = np.sqrt(mean_squared_error(y_test_churn_inv, y_pred_churn_inv))
mae_churn = mean_absolute_error(y_test_churn_inv, y_pred_churn_inv)

print("LSTM - Churn Rate: RMSE =", rmse_churn, "MAE =", mae_churn)

# Compare with ARIMA
model_arima_churn = ARIMA(df['Churn_Rate'][:train_size], order=(1, 0, 1)).fit()
arima_pred_churn = model_arima_churn.forecast(steps=len(df) - train_size)
mae_arima_churn = mean_absolute_error(df['Churn_Rate'][train_size:], arima_pred_churn)
print("ARIMA(1,0,1) - Churn Rate MAE =", mae_arima_churn)
```

Conclusion:

- **LSTM outperforms ARIMA** in Churn Rate prediction.
- Ideal for time series with **irregular spikes**, **anomalies**, or **complex seasonality**.

LSTM Churn Rate Forecast



LSTM Churn Rate Forecast

Forecast window: 14 days (Mar 9 to Mar 22, 2025).

Input: Last 7 days of churn data used in rolling fashion to forecast next day.

Predicted churn range: ~0.1779 to ~0.2189.

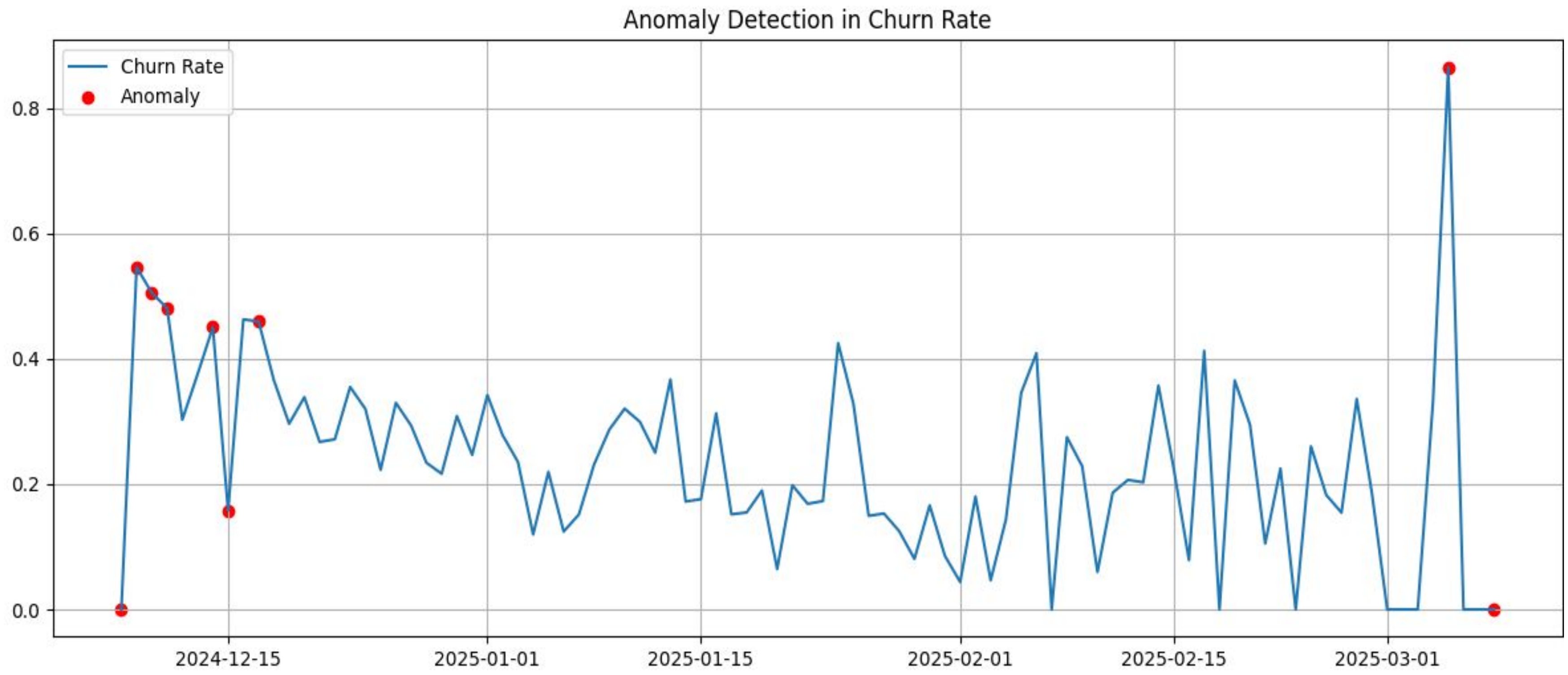
Forecast pattern:

- Starts at 0.1962, peaks around Mar 11 (0.2184).
- Slight dip on Mar 13 (0.1779), then stabilizes near 0.206.

Model trait:

- Produces smooth, steady predictions, ideal for trend-level forecasting.
- Misses abrupt spikes but maintains consistency in flat or moderate churn scenarios.

ANOMALY PLOT



Anomaly Detection in Churn Rate

- Technique Used: Isolation Forest applied to `Churn_Rate` feature.
- Anomalies Detected: Marked as red dots — these represent unusually high or low churn spikes.
- Notable Insights:
  - Sudden spike around March 2025 is flagged as a major anomaly.
  - Early high churn days in December 2024 also identified as anomalies.
- Purpose:
  - Helps in identifying user behavior shifts or platform issues.
  - Supports proactive response by marketing/ops to prevent churn spikes.
- Business Value: Enables early intervention and improves customer retention strategy.

COMPARISON

LSTM Performance:

- MAE: 0.1651
- RMSE: 0.2250
- Predicted smooth curves, struggled with sudden spikes.
- Forecast captured general trend

ARIMA Performance:

- MAE: 0.1753
- Slightly higher error than LSTM.
- Forecast was flat and less responsive, even on differenced series.
- Captured seasonality poorly, though residuals were somewhat stable.

CONCLUSION

LSTM performs better overall

- Lower MAE and RMSE.
- Better alignment with actual churn patterns in test set.
- More adaptive in non-linear behavior, which suits churn dynamics
- Better alignment with actual fluctuations in user behavior, despite some deviations.

Future Scope

- Automated Decision Triggers

Connect the dashboard to backend systems so that, when a churn risk or anomaly is detected, it can trigger actions automatically, like:

- Sending discount emails
- Flagging accounts for review
- Notifying customer support
- Mobile & Multi-Platform Integration

Make the dashboard available via mobile devices or API access so internal teams can monitor things on the go.

Allow different teams (product, marketing, customer success) to view tailored versions of the dashboard.