Forecasting Transaction Volumes on the XRP Ledger Using Machine Learning

Alexandre Amalric
XRPL Commons - Ripple

January 2025

Abstract

Building upon the 13 Stock Predictor Big Data project, this paper transitions to a more complex challenge of forecasting transaction volumes on the XRP Ledger. The focus is on developing predictive models capable of capturing intricate patterns in blockchain transaction activity. By leveraging time-series models such as RNNs, GRUs, LSTMs, and Transformers, we aim to predict transaction volume trends based on historical data and external features.

The project includes extensive data preparation: aggregating, engineering, and normalizing large datasets from the XRP Ledger, and integrating auxiliary data such as market indicators and social media sentiment. A combination of supervised, unsupervised, and reinforcement learning techniques is applied to adapt the models to dynamic transaction behaviors.

The goal is to provide a scalable, interpretable framework for transaction volume forecasting, offering actionable insights for blockchain validators, investors, and network monitors. This research supports enhanced resource allocation, risk management, and operational efficiency within the XRPL ecosystem.

1 Introduction

1.1 Background

The XRP Ledger (XRPL) is a high-performance blockchain optimized for fast and low-cost cross-border payments. As blockchain networks grow in usage and complexity, understanding and predicting transaction volumes has become critical. Accurate forecasting of transaction volumes on XRPL can enhance operational efficiency, inform resource management, and provide insights for financial applications and investment strategies. Predicting transaction volumes enables stakeholders to anticipate demand, optimize resource allocation, and adapt to fluctuations in network activity, which has implications for performance and cost management on the blockchain.

1.2 Objective and Significance

This study aims to develop a comprehensive machine learning framework for predicting transaction volumes on the XRP Ledger using advanced time-series forecasting techniques. We explore the application of Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Transformers to analyze both historical and auxiliary data, creating a robust predictive model. The need for precise volume forecasting extends across various stakeholder groups:

- **Validators and Network Operators**: Validators play a critical role in the XRPL by maintaining the integrity of the blockchain and processing transactions. Accurate volume forecasting helps validators prepare for periods of high demand, ensuring that they allocate sufficient computational resources to handle transaction peaks without compromising network performance. Volume forecasts allow for better network scaling and load balancing, thereby reducing latency and enhancing the overall user experience.
- **Investors and Financial Analysts**: Investors closely monitor transaction volumes as an indicator of network usage, adoption, and user engagement. High transaction volumes are often associated with increased utility and, consequently, network value, making volume forecasts an

important metric for investment decisions. Predicting transaction trends enables investors to anticipate shifts in network activity, manage risk, and make informed decisions on token holdings, thereby supporting more strategic portfolio management.

- **Blockchain Monitoring and Compliance Teams**: Entities responsible for monitoring the XRPL, including compliance officers and security analysts, rely on transaction volume trends to detect anomalies and ensure the network functions smoothly. Sudden spikes or drops in transaction volume can indicate security threats, regulatory breaches, or operational issues. Forecasting volume trends aids in proactive monitoring and enhances the capacity to address irregularities promptly, supporting blockchain integrity and compliance with regulations.
- **Network Developers and Infrastructure Providers**: Developers and service providers who build applications on top of the XRPL or provide blockchain infrastructure benefit from volume forecasts for capacity planning and service optimization. Forecasting transaction volumes helps developers anticipate periods of increased usage, allowing them to adapt service capacities and scale resources as needed to avoid downtime or degraded performance during peak periods.
- **Market Participants and Arbitrageurs**: Transaction volumes can impact the liquidity and price of assets traded on the XRPL, making volume forecasting valuable for traders and arbitrageurs who rely on high liquidity for efficient market operations. Volume trends help these participants anticipate market movements, detect price discrepancies, and execute trades at optimal times.

1.3 Challenges in Transaction Volume Forecasting

Forecasting transaction volumes on a blockchain like the XRP Ledger poses several challenges due to the dynamic and multifaceted nature of blockchain ecosystems:

- **High Volatility and External Influences**: Transaction volumes are influenced by a variety of external factors, including market sentiment, regulatory news, and global economic events. The high volatility in transaction activity necessitates robust models that can capture the effect of these variables on transaction volumes.
- **Complex Temporal Dependencies**: Blockchain transaction patterns exhibit complex temporal dependencies, with volumes fluctuating based on time of day, day of the week, and seasonality. Capturing these dependencies requires advanced models like RNNs and Transformers, which can learn from sequential data and identify cyclic patterns that affect transaction behavior.
- **Data Heterogeneity and Scalability**: The XRPL encompasses a vast amount of transaction types (e.g., payments, trustlines, smart contracts) and token-specific data, which adds complexity to the modeling process. Additionally, the large size of historical transaction data requires scalable data processing and model training methods to ensure efficient and accurate predictions.
- **Anomaly Detection and Adaptability**: Transaction volumes are susceptible to unexpected surges or declines due to sudden market events or coordinated activities, such as airdrops or large fund transfers. Effective forecasting models must not only capture regular trends but also adapt to anomalies, enabling the model to respond to outliers without overfitting.

By addressing these challenges, our research aims to provide a forecasting model that can enhance decision-making and operational efficiency across the XRPL ecosystem. Through extensive data preparation, model selection, and evaluation, we develop a scalable solution that leverages both on-chain and off-chain data to capture the complex patterns of transaction volume on the XRP Ledger.

2 Data Preparation

2.1 Data Access and Aggregation

The XRP Ledger dataset, consisting of ledger.db (32 GB) and transaction.db (8.2 TB), provides extensive transaction and account information. Given the large size of these datasets, we adopt a phased approach, beginning with data from external APIs like CoinGecko `https://www.coingecko.com` for initial model training and then refining our model with more granular data from the XRP Ledger databases.

- **API Data Retrieval**: To establish a baseline and facilitate efficient data handling, we utilize the CoinGecko API to retrieve historical transaction volume data for major tokens on the XRP Ledger. This API provides transaction volumes at various intervals, allowing us to construct initial time-series data for model training.
- **Additional External Data**: We incorporate various external data sources to enhance our model's predictive power. These include:
  - **Market Indicators**: Real-time and historical XRP price, trading volume, and volatility metrics sourced from CoinGecko and other financial platforms, as market trends often correlate with on-chain transaction activity.
  - **Social Media Sentiment**: Using social media data from platforms like Twitter and Reddit, we track mentions and sentiment around XRP, which can act as an early indicator of trading interest and potential transaction volume spikes.
  - **Global Economic Data**: Key macroeconomic indicators (e.g., inflation rates, interest rates) are also included, as they may indirectly influence blockchain activity and transaction volumes.
- **Database Schema Exploration**: Following initial model development with API and external data, we explore ledger.db and transaction.db schemas for relevant fields, such as LedgerSeq, ClosingTime, TotalCoins, and transaction-specific details like amounts and transaction types, to enhance the dataset.
- **Data Aggregation**: For deeper analysis, transaction data is aggregated by daily or hourly intervals, with total transaction volumes and frequencies calculated per interval to capture broader volume trends.

2.2 Feature Engineering

To improve model performance and capture the temporal patterns in transaction volume, we develop a set of features that represent cyclical and sequential behaviors within the data.

- **Lagged Volume Features**: Previous transaction volumes are included as lagged features to capture autocorrelation patterns and inform the model of past trends.
- **Time-based Features**: Temporal indicators, such as day of the week, month, and hour, are added to capture cyclic behaviors and potential seasonal effects within transaction volumes.
- **Market and Social Sentiment Features**: We encode market data (e.g., XRP price, trading volume) and social media sentiment metrics as additional features to help the model account for external influences on transaction volume.
- **Token and Transaction Types**: To account for volume variations driven by different asset types and transaction categories, we perform categorical encoding of token-specific transactions and transaction types (e.g., trustlines, smart contracts).

This approach allows for initial model development with manageable API data and enhanced features, followed by refinement using the complete XRP Ledger datasets. By integrating both on-chain and off-chain data, we aim to improve the accuracy and robustness of our transaction volume forecasts.

3 Data Preprocessing

3.1 Normalization

Transaction volumes exhibit high variability, so we apply min-max normalization to scale values between 0 and 1, preserving temporal trends for model training:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

This scaling standardizes input ranges, improving model convergence during training.

3.2 Handling Missing Values

Occasional gaps in transaction data are interpolated using forward filling, ensuring continuity in time-series input sequences without introducing bias in volume patterns.

4 Model Selection Roadmap

To effectively forecast transaction volumes on the XRP Ledger, we progress through a structured roadmap of increasingly complex models, beginning with simpler neural networks and gradually incorporating advanced techniques and additional features. This roadmap allows us to compare models iteratively, analyzing the impact of added complexity and external data on prediction accuracy.

4.1 Step 1: Baseline Supervised Models with Simple Recurrent Neural Networks

Objective: Establish a baseline for volume prediction using basic Recurrent Neural Networks (RNNs), providing a foundation for measuring improvements as we increase model complexity.

Data Source: We use the `ledger.db` dataset, specifically extracting the following fields:

- `LedgerSeq`: Sequential ID of the ledger.
- `ClosingTime`: Timestamp marking the end of each ledger.
- `TotalCoins`: Total number of coins in circulation, which provides context for network activity.

Feature Engineering: To construct the initial dataset, we aggregate transaction volumes by hourly intervals, using simple time-based features (e.g., day of the week) to capture cyclic patterns.

Model: Recurrent Neural Network (RNN) RNNs are supervised learning models that predict future values based on past data, retaining information through hidden states across time steps. However, standard RNNs face challenges with vanishing gradients, limiting their ability to capture long-term dependencies.

Mathematical Formulation: Given a sequence of transaction volumes $V = \{v_1, v_2, \ldots, v_t\}$ over time t, the RNN learns a function $f : V \to v_{t+1}$ where v_{t+1} is the next predicted transaction volume:

$$h_t = \sigma(W_h h_{t-1} + W_x v_t)$$

where h_t is the hidden state at time t, σ is an activation function, and W_h and W_x are weight matrices.

Expected Output: Predicted transaction volume for the next time interval. This model serves as a baseline, allowing us to compare future, more complex models against this initial prediction accuracy.

—

4.2 Step 2: Intermediate Recurrent Models with Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM)

Objective: Improve the model's ability to capture longer-term dependencies in transaction volumes by using GRU and LSTM networks, which address vanishing gradient issues.

Data Source: We continue using the `ledger.db` fields, now adding external data from CoinGecko, including:

- `Price of XRP`: Historical price of XRP, retrieved via the CoinGecko API.
- `Market Sentiment`: Social media sentiment scores for XRP as a proxy for external interest.

Feature Engineering: Lagged transaction volumes are added as features to capture autocorrelation. Additionally, we encode the day of the week and hour to capture time-based trends and align CoinGecko price and sentiment data to each ledger's timestamp.

Models: GRU and LSTM

- **GRU (Gated Recurrent Unit)**: GRUs use gating mechanisms to control information flow, helping to retain or discard information over time, thereby capturing medium-term patterns.
- **LSTM (Long Short-Term Memory)**: LSTMs improve upon GRUs with an additional memory cell, enabling the model to retain relevant information over longer sequences, capturing seasonal or weekly transaction volume cycles.

Mathematical Formulation (LSTM): For each time step t, the LSTM has cell state c_t and hidden state h_t, with updates given by:

$$f_t = \sigma(W_f \cdot [h_{t-1}, v_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, v_t] + b_i)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, v_t] + b_o)$$

where f_t, i_t, and o_t are the forget, input, and output gates respectively.

Expected Output: Predicted transaction volume for the next interval, with improved accuracy over the RNN model, particularly in cases where volume changes are seasonal or cyclical.

—

4.3 Step 3: Advanced Temporal Models with Temporal Convolutional Networks (TCN)

Objective: Capture complex dependencies across long sequences without the vanishing gradient problem, improving performance on high-dimensional transaction data.

Data Source: We use both `ledger.db` and `transactions.db`, including:

- `TransactionType`: Categorized from `transactions.db` to capture volume drivers by type (e.g., payments, trustlines).
- `LedgerSeq` and `TotalCoins` from `ledger.db` for context.

Feature Engineering: We aggregate volume by transaction type (e.g., payment, trustline) and apply one-hot encoding. TCNs can also process auxiliary data (e.g., time, price) alongside volume data.

Model: Temporal Convolutional Network (TCN) TCNs are convolutional networks designed for time-series data, allowing the model to process entire sequences at once using causal convolutions, thus avoiding the limitations of recurrent layers.

Mathematical Formulation: Given a sequence V, TCNs apply a convolutional filter f over causal steps:

$$h_t = f(h_{t-1}, v_t, \ldots, v_{t-k})$$

where k is the receptive field of the filter.

Expected Output: Predicted transaction volume, with improved handling of high-dimensional data and reduced computational requirements.

—

4.4 Step 4: Transformer-Based Models for Long-Sequence Forecasting

Objective: Leverage self-attention to capture dependencies across long time horizons, even when those dependencies span weeks or months.

Data Source: In addition to `ledger.db` and `transactions.db`, we incorporate CoinGecko data for external indicators (XRP price, trading volume, sentiment), making the dataset multivariate.

Feature Engineering: Include time-based and token-specific features with position encoding to help the Transformer model retain sequential information.

Models: Temporal Fusion Transformer (TFT) and Informer

- **Temporal Fusion Transformer (TFT)**: This Transformer variant captures variable importance dynamically, adapting to shifts in relevant features.
- **Informer**: Optimized for long-sequence forecasting, using sparse attention to improve efficiency for large datasets.

Mathematical Formulation (Attention Mechanism): For query Q, key K, and value V, attention is computed as:

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

where d_k is the dimension of the keys.

Expected Output: Long-sequence transaction volume predictions with high interpretability, as the model can indicate which features and time intervals most impact the prediction.

—

4.5 Step 5: Probabilistic Models and Reinforcement Learning for Adaptive Forecasting

Objective: Introduce uncertainty estimation and adapt the model dynamically to improve predictive robustness in volatile conditions.

Data Source: Full dataset, including all features from previous steps.

Models: Bayesian LSTM and Policy Gradient Reinforcement Learning

- **Bayesian LSTM**: Provides confidence intervals around predictions.
- **Policy Gradient Reinforcement Learning**: Adapts forecasting strategy based on error feedback.

Expected Output: Probabilistic forecast with confidence intervals, enabling better risk management and adaptive strategy based on past prediction errors.

—

5 Experiment Setup

5.1 Data Splitting

The dataset is split chronologically, with 80% for training and 20% for testing, ensuring the model generalizes to future volume trends.

5.2 Hyperparameters and Training

For each model, we tune hyperparameters such as the number of epochs, learning rate, and batch size. Table 1 lists the settings used.

Hyperparameter	Value
Epochs	100
Learning Rate	0.001
Batch Size	64

Table 1: Hyperparameters for Time-Series Models

5.3 Evaluation Metrics

To assess model performance, we use:

- **Mean Absolute Error (MAE)**: Measures the average absolute error between predicted and actual values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

- **Root Mean Squared Error (RMSE)**: Provides insight into error magnitude, penalizing larger deviations.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

6 Results and Analysis

6.1 Model Performance

Each model's performance was evaluated on the test set using RMSE, MAE, MAPE, sMAPE, and R^2. The results are presented in Table 2.

Model	RMSE	MAE	MAPE	sMAPE	R^2
XGBoost	0.3108	0.2938	260.7335	89.3098	-11.0245
Random Forest	0.1801	0.1652	161.8618	63.7418	-3.0407
DeepAR	**0.0990**	**0.0630**	**54.3235**	**32.6781**	-0.2209
Simple LSTM	0.0995	0.0740	76.2526	35.3664	-0.2322

Table 2: Model performance on the test set for transaction volume forecasting.

6.2 Error Analysis

The results show a significant difference in performance across models:

- **DeepAR**: Achieved the lowest RMSE and MAE, indicating its ability to capture patterns in transaction volume effectively. However, the R^2 value is negative, reflecting poor trend alignment.
- **XGBoost**: Performed the worst, with the highest RMSE and MAE and a severely negative $R^2 = -11.0245$, highlighting its limitations for time-series forecasting.
- **Random Forest**: Improved over XGBoost but still struggled with capturing the inherent patterns in transaction data.
- **Simple LSTM**: Comparable to DeepAR in RMSE but performed slightly worse in sMAPE and MAE.

6.3 Understanding Negative R^2

A negative R^2 indicates that the model's predictions are worse than simply using the mean of the observed values as a constant predictor. The causes of this include:

- **High Volatility:** Blockchain transaction volumes are highly volatile, making it difficult for models like XGBoost and Random Forest to generalize well.

Model	RMSE	MAE	MAPE	sMAPE	R2
XGBoost	0.3108	0.2938	260.7335	89.3098	-11.0245
RandomForest	0.1801	0.1652	161.8618	63.7418	-3.0407
DeepAR	0.0990	0.0630	54.3235	32.6781	-0.2209
SimpleLSTM	0.0995	0.0740	76.2526	35.3664	-0.2322

Figure 1: Performance metrics comparison (RMSE, MAE, MAPE, sMAPE, R^2).

- **Mean Reversion:** Models often converge toward the mean of the data, failing to capture spikes or drops in transaction activity.
- **Overfitting on Training Data:** Tree-based models may overfit the training data while failing to generalize to the test set, leading to poor predictions.
- **Inability to Capture Sequential Dependencies:** Models like XGBoost and Random Forest lack mechanisms to account for time-series autocorrelations and seasonality, unlike RNN-based models.

6.4 Interpretability

Despite the challenges, DeepAR and LSTM models exhibit better interpretability in handling time-series data. By capturing temporal dependencies, these models slightly outperform tree-based methods, which struggle in the presence of high volatility and noise.

7 Conclusion and Future Work

7.1 Summary of Findings

This study highlights the limitations of machine learning models in forecasting highly volatile transaction volumes on the XRP Ledger. Key takeaways include:

- **DeepAR** achieved the best RMSE and MAE but still resulted in negative R^2, emphasizing the difficulty of modeling such complex time-series data.
- Tree-based models like XGBoost and Random Forest performed poorly due to their inability to capture sequential patterns.
- Negative R^2 reflects the significant challenges in aligning predictions with actual trends in the data.

7.2 Future Directions

To address these challenges, future research should:

- Explore hybrid architectures (e.g., CNN-Transformers) to capture both short- and long-term dependencies.
- Incorporate external features such as market conditions or social media sentiment for improved context.
- Enhance preprocessing techniques to better capture trends and seasonal components in transaction data.