Tunis Business School

Department of Finance

FIN 460 – Dynamic Asset Pricing Theory

A Real-Time Intrinsic Valuation Model for the Tunisian Dinar (TND)

Authors:

Mustapha Aziz Laroussi
Salma Regaieg
Yassine Chakroun
Mohamed Mechri

Instructor: Dr. Eymen Errais

Submission Date: May 31, 2025

Abstract

We propose a real-time intrinsic valuation model for the Tunisian Dinar (TND), tailored to bridge the persistent gap between the Central Bank's fixing and the interbank exchange rate. Our framework integrates global currency fundamentals and local market frictions through a two-layer stochastic model. The first layer is a dynamic basket-based baseline, modeling USD/TND log returns as a time-varying linear combination of major FX pairs:

$$\Delta \log(F_t) = \mathbf{w}_t^\top \Delta \log(\mathbf{X}_t),$$

where the weight vector $\mathbf{w}_t$ evolves stochastically via a state-space process estimated by a Kalman Filter. The second layer introduces a latent spread component s_t, representing intraday liquidity pressures, jointly modeled with the weights in a Double Kalman Filter:

$$\Delta \log(F_t) = \mathbf{w}_t^\top \Delta \log(\mathbf{X}_t) + s_t + \varepsilon_t.$$

To further enhance accuracy, a neural network residual corrector is trained on lagged prediction errors and filtered states, forming a hybrid forecasting structure. The final model outperforms classical regression and baseline machine learning benchmarks, achieving a MAPE of 0.1452%, RMSE of 0.0063, and correlation of 0.9919 with the interbank rate. This layered, adaptive system demonstrates both statistical robustness and practical utility for nowcasting in FX markets affected by dual global-local dynamics.

1 Introduction

The valuation of a national currency is a very important aspect of an economy that is affected by many factors including the trading balance of a country. In the case of Tunisia, a net importer, the Central Bank of Tunisia (BCT) plays a pivotal role in determining the official exchange rate of the Tunisian Dinar (TND), particularly against the U.S. Dollar (USD). This rate, published twice daily as the official fixing, is set by market conditions and is affected by them. It is often conceptualized as a weighted function of major currency pairs such as EUR/USD, GBP/USD, and USD/JPY.

While this basket-based approach provides a useful theoretical benchmark, the actual TND rate observed in Tunisia's foreign exchange (FX) market often deviates from the fixing. These deviations are driven by many factors, primarily local liquidity pressures. A key symptom of this phenomenon is the persistent spread between the official fixing and the interbank (IB) USD/TND exchange rate, which reflects the actual rate at which Tunisian banks trade foreign currency. However, the IB rate is only published with a one-day lag, making it inadequate for real-time valuation and trading decisions.

Since we can't rely on the interbank rates, discovering market realities requires a more responsive and dynamic valuation framework, which makes a strong case for constructing a model that captures both international currency fundamentals and real-time local market frictions. Such a model would provide a more accurate estimate of the TND intrinsic value, serving as a better reference point for FX traders and policymakers.

While the assignment proposes a model with a static component and a stochastic one, we figured that it is not wise to assume that the weights are static, and, after consulting with Mr. Erraies, we decided to take it a step further: we assume that the weights in the first component are stochastic, allowing the baseline to evolve dynamically with global market movements. This assumption leads to a more flexible and accurate valuation framework.

That being said, we propose a real-time intrinsic valuation model for the Tunisian Dinar composed of two interacting stochastic components. The first is a **basket-based baseline**, which estimates the fair value of the USD/TND exchange rate based on a weighted combination of major currency pairs (EUR/USD, GBP/USD, and USD/JPY). The second component is a **stochastic spread adjustment**, which accounts for local market conditions such as FX liquidity shortages and intraday supply-demand imbalances. This spread represents the deviation between the observed market rate and the basket-implied value.

By combining these two elements—a stochastic, globally-informed baseline and a stochastic, locally-informed spread—our model produces a continuously updating estimate of the intrinsic USD/TND exchange rate.

2 Model Framework Design

This section presents the design of a hybrid model aimed at estimating the real-time intrinsic value of the USD/TND exchange rate. The model is structured in two key components:

1. A basket-based baseline estimation using major currency pairs with stochastic weights.
2. A residual correction mechanism that captures local liquidity deviations.

2.1 Data Transformation and Preprocessing

All financial time series (exchange rates and USD/TND fixing/interbank rates) are first transformed into **log returns** to ensure stationarity, symmetry, and approximate normality. Given a price series P_t, the log return is computed as:

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right)$$

After transformation, all series are **standardized** (zero mean and unit variance) to ensure compatibility across variables during model training and filtering.

2.2 Basket-Based Baseline Estimation

The baseline estimation models the intrinsic value of USD/TND as a dynamic combination of returns from a currency basket:

$$P_t^{\text{basket}} = P_t^{\text{fixing}} \cdot (w_{1,t} \cdot \Delta\text{EUR/USD}_t + w_{2,t} \cdot \Delta\text{GBP/USD}_t + w_{3,t} \cdot \Delta\text{USD/JPY}_t)$$

Where:

- P_t^{fixing}: The most recent BCT fixing rate for USD/TND.

- ΔX_t: Log return of currency pair X at time t.
- $w_{i,t}$: Stochastic weights, estimated dynamically.

The weight vector $\mathbf{w}_t = [w_{1,t}, w_{2,t}, w_{3,t}]^\top$ is assumed to evolve over time as a stochastic process:

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, Q)$$

This formulation allows the model to adapt to changing relationships between the USD/TND exchange rate and the basket currencies.

2.3 Residual Correction via Liquidity Spread

To account for deviations between the modeled basket value and the actual interbank rate, we model a stochastic spread term S_t:

$$S_t = \hat{P}_t^{\text{IB}} - P_t^{\text{basket}}$$

Where $\hat{P}_t^{\text{IB}}$ is a predicted or observed interbank rate. Alternatively, S_t can be directly modeled as a mean-reverting process:

$$S_t = \rho S_{t-1} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

2.4 Final Model Formulation

The real-time intrinsic value of USD/TND is computed as:

$$\boxed{\widehat{P}_t = P_t^{\text{fixing}} \cdot \left(\sum_{i=1}^{3} w_{i,t} \cdot \Delta C_{i,t} \right) + S_t}$$

- $\Delta C_{i,t}$: Log return of the i^{th} currency pair at time t.
- $w_{i,t}$: Kalman-filtered weight at time t.
- S_t: Liquidity-driven stochastic spread.

2.5 Real-Time Operation

In real-time deployment:

1. Currency pair log returns are observed.

2. The Kalman filter updates weight and S_t estimates based on previous errors.

3. A neural network further adjusts S_t for more accuracy.

4. The final value $\widehat{P}_t$ is computed and can be used as a nowcast for the interbank rate.

This modular structure allows each component to be updated independently and facilitates real-time forecasting under varying market conditions.

3 Data Transformation – Testing & Log Returns

The initial phase of testing involved preparing and analyzing the raw data provided before proceeding with weight estimation and modeling. This critical step focuses on transforming the data and conducting statistical tests to understand its properties, particularly stationarity and autocorrelation, which are important for time-series modeling.

3.1 Log Returns

We decided to utilize log returns for the currency pair movements and the USD/TND Fixing rate, rather than simple percentage returns. The choice is based on three main properties:

1. **Symmetry:** Log returns treat upside and downside movements equally and present them equidistantly on the log scale.

2. **Normality:** Daily log returns tend to be closer to a normal distribution, beneficial for many modeling techniques, including machine learning.

3. **Stationarity:** Computing log returns improves the stationarity of the time series.

3.2 Normalization of the FX Rates

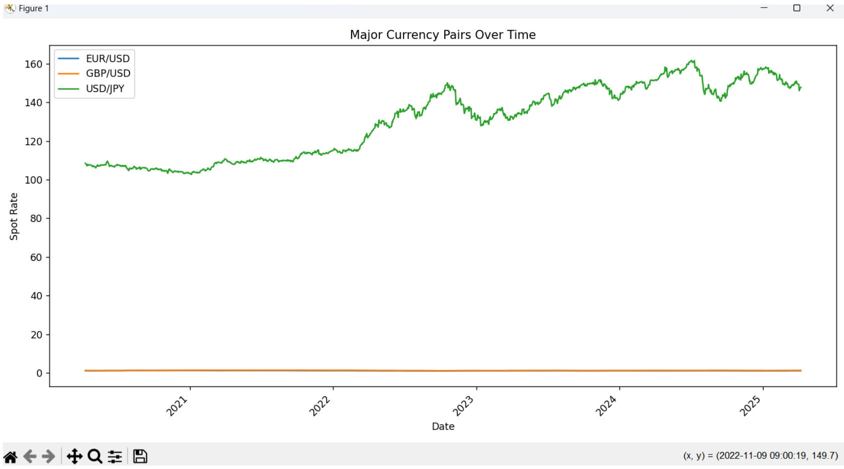


Figure 1: FX rates before normalization

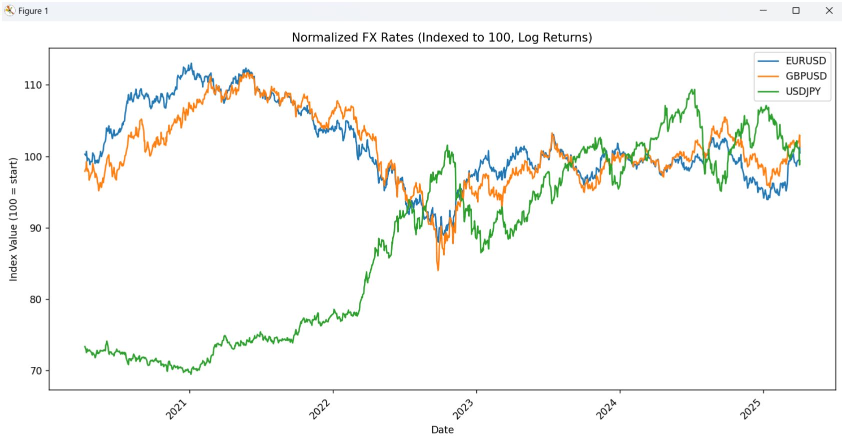


Figure 2: FX rates after normalization

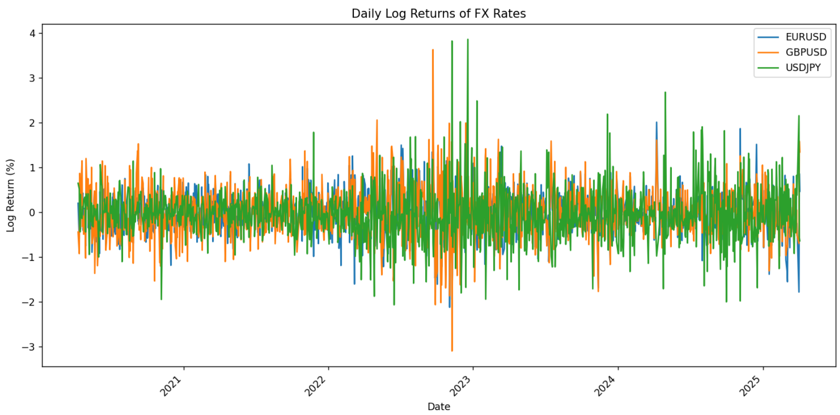


Figure 3: Log Returns

After cleaning missing entries and normalizing the price series, we computed the daily log returns for each series (EURUSD, GBPUSD, USDJPY, Fixing_Mid, IB_USD) as:

$$r_t = \ln\Big(P_t\Big) - \ln\Big(P_{t-1}\Big).$$

3.3 Statistical Testing: Stationarity and Autocorrelation

To formally assess the properties of the transformed data, we performed:

- **Augmented Dickey–Fuller (ADF) Test:** Tests the null hypothesis of a unit root (non-stationarity) in each log-return series.
- **Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) Plots:** Measure correlation of each series with its own lagged values.

Table 1: ADF Test Results for Log Returns

Series	ADF Statistic	Crit. Value (1%)	Crit. Value (5%)	Crit. Value (10%)
Fixing_Mid	-35.2679	-3.4356	-2.8638	-2.5680
IB_USD	-33.2189	-3.4356	-2.8638	-2.5680
EURUSD	-26.1741	-3.4356	-2.8638	-2.5680
GBPUSD	-26.3589	-3.4356	-2.8638	-2.5680
USDJPY	-17.7293	-3.4356	-2.8638	-2.5680

In all cases, the p-values are effectively zero, allowing rejection of the null hypothesis of non-stationarity. Thus, the log returns of all series can be considered stationary.

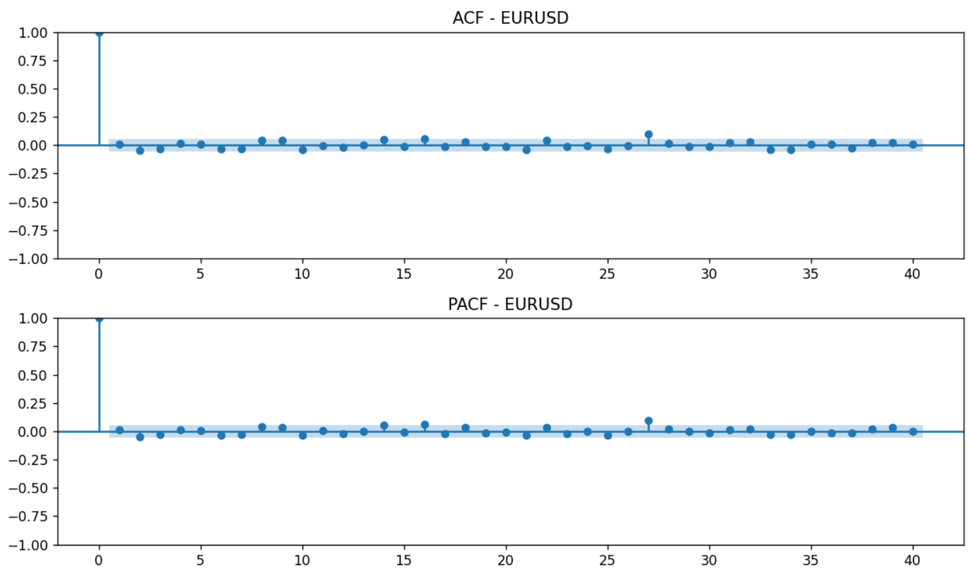


Figure 4: ACF and PACF plots for EURUSD log returns

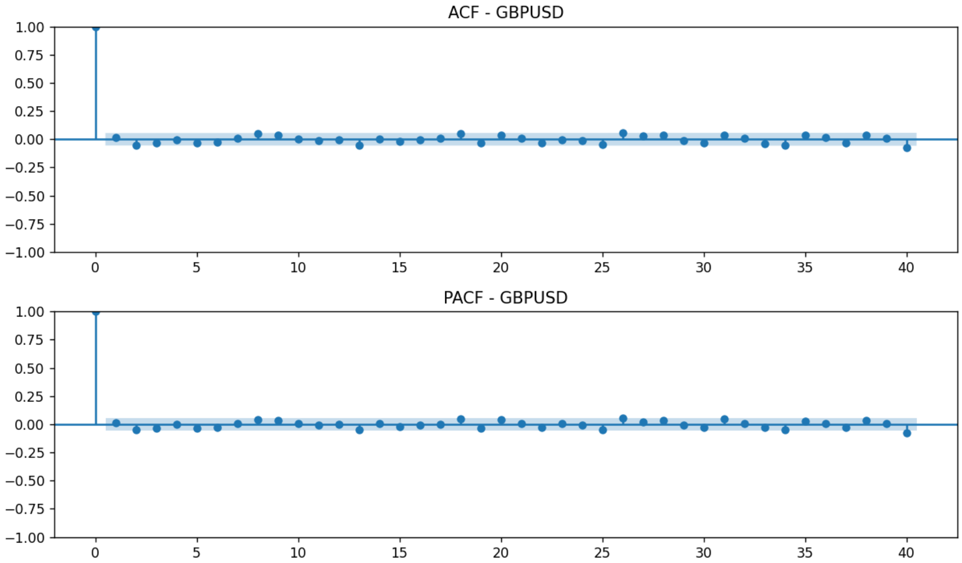


Figure 5: ACF and PACF plots for GBPUSD log returns

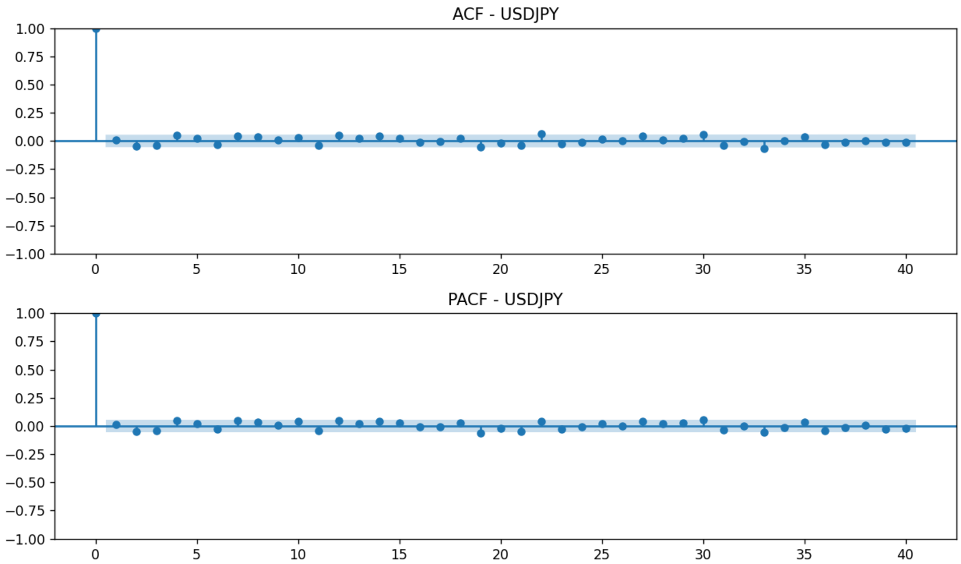


Figure 6: ACF and PACF plots for JPYUSD log returns

The ACF and PACF in plots 4, 5 and 6 show no significant autocorrelation beyond lag 1. The correlation spikes at lag 0 (which is always 1) and then quickly falls within the confidence bounds for subsequent lags.

Conclusion: Based on the ADF tests and the ACF/PACF analysis, all log-return series are stationary with no significant autocorrelation beyond the first lag, making them suitable for subsequent time-series modeling techniques.

4 First Approach: Classic Regression

A core assumption we made is that the weights are stochastic and vary over time, and the relationship between the currencies and USD/TND is not necessarily linear. Given these assumptions, traditional regression methods, which estimate constant weights, are suboptimal for achieving the real-time accuracy targeted. However, we still conducted tests with this approach to serve as a baseline for comparison.

4.1 Methodology

- Perform ordinary least squares (OLS) regression using `LinearRegression` from scikit-learn.
- Target variable: daily log returns of the USD/TND fixing, r_t^{Fixing}.
- Features: daily log returns of EUR/USD, GBP/USD, and USD/JPY, denoted $r_t^{\text{EUR/USD}}$, $r_t^{\text{GBP/USD}}$, $r_t^{\text{USD/JPY}}$.
- Split data into an 80% training set and a 20% test set.
- Fit OLS on the training set, then predict log returns on the test set using the fitted, constant weights.
- Reconstruct predicted fixing levels via

$$\widehat{P}_t^{\text{Fixing}} = P_{t-1}^{\text{Fixing}} \exp\Big(\widehat{r}_t^{\text{Fixing}}\Big).$$

4.2 Regression Equation

The estimated model has the form

$$r_t^{\text{Fixing}} = \beta_{\text{EUR/USD}}\, r_t^{\text{EUR/USD}} + \beta_{\text{GBP/USD}}\, r_t^{\text{GBP/USD}} + \beta_{\text{USD/JPY}}\, r_t^{\text{USD/JPY}} + \varepsilon_t.$$

Table 2: Estimated Constant Weights and Out-of-Sample Errors

Parameter	Value
Estimated weight EUR/USD	-0.3498
Estimated weight GBP/USD	-0.0358
Estimated weight USD/JPY	0.0207
Out-of-sample RMSE	0.009795
Out-of-sample MAPE (%)	0.217
R-square	0.9606

4.3 Results

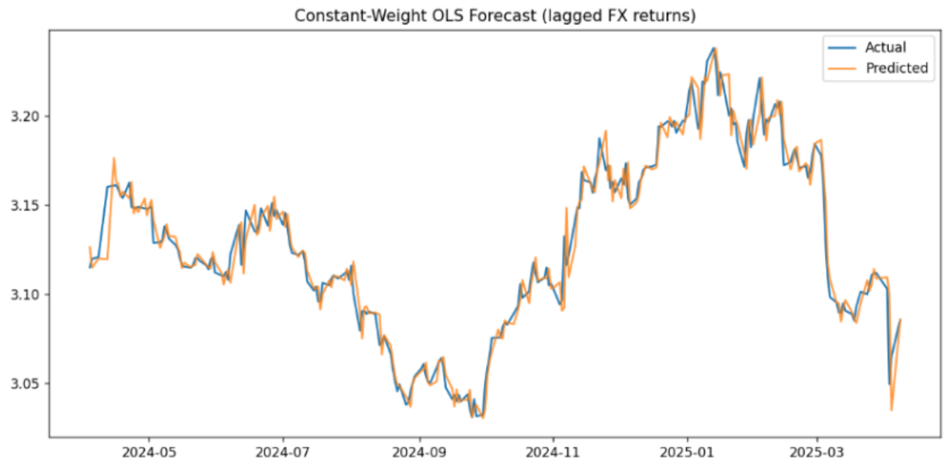


Figure 7: Actual vs. predicted USD/TND fixing levels using constant-weight regression

The constant-weight regression provides a simple baseline that performs respectably, though dynamic models (discussed in later sections) offer improved adaptation to volatility and market shifts.

5 Second Approach: Machine Learning Model

The initial formula is structured as follows:

$$\text{USD/TND} = (\text{Latest fixing}) \times \Big(w_1 \times \%\Delta\text{EUR/USD} + w_2 \times \%\Delta\text{GBP/USD} + w_3 \times \%\Delta\text{USD/JPY}\Big).$$

As we rejected the traditional regression methods, machine learning was identified as the next optimal method to pursue. A crucial preliminary step was the decision to use log returns instead of percentage returns. This choice was based on the properties of log returns.

The relationship was thus reformulated in terms of log returns:

$$\log(Fixing_t) - \log(Fixing_{t-1}) = f(\text{time-varying log FX returns}).$$

We then aimed to estimate these desired time-varying weights using a machine learning model. Our selection went to the `SGDRegressor` model from the scikit-learn library. Our intention was to predict the daily log return of the USD/TND fixing ($\log(Fixing_t) - \log(Fixing_{t-1})$) using the daily log returns of EUR/USD, GBP/USD, and USD/JPY as features.

To capture the time-varying nature of the weights, the `SGDRegressor` was intended to be updated iteratively day by day using the `partial_fit` method. This simulates a learning process where the model adapts as new data arrives. The implementation involved the following steps:

1. Loading the data for the Fixing, IB rates, and major currencies from our Excel file and merging them based on date.

2. Computing daily log returns for the relevant time series.

3. Standardizing the FX log returns and Fixing log return.

4. Looping through the data, using with the previous day's scaled data to update the model's weights.

5. Predicting the current day's scaled log return using the updated model, then inverse-transforming it to get the actual log return prediction.

6. Storing the model's estimated weights at each step.

7. Reconstructing the predicted Fixing level by multiplying the previous day's actual Fixing level by the exponential of the predicted log return.

The results of this first approach were the following:

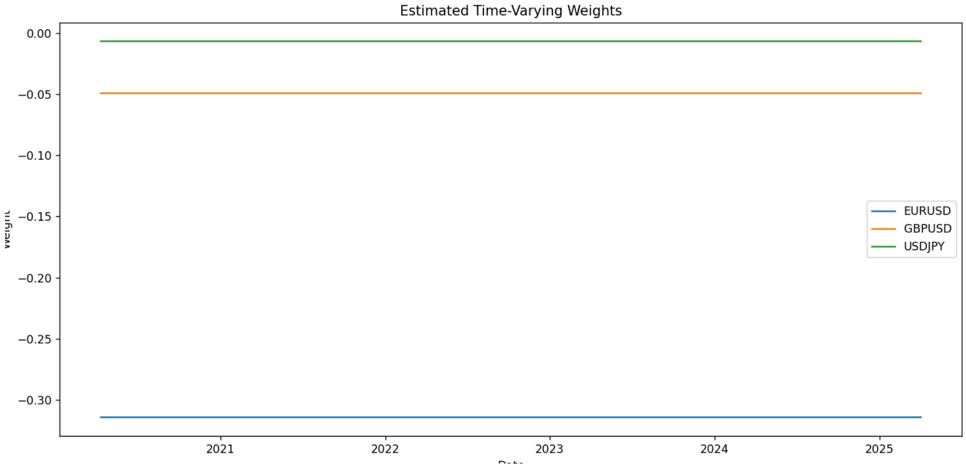


Figure 8: Estimated Time-Varying Weights

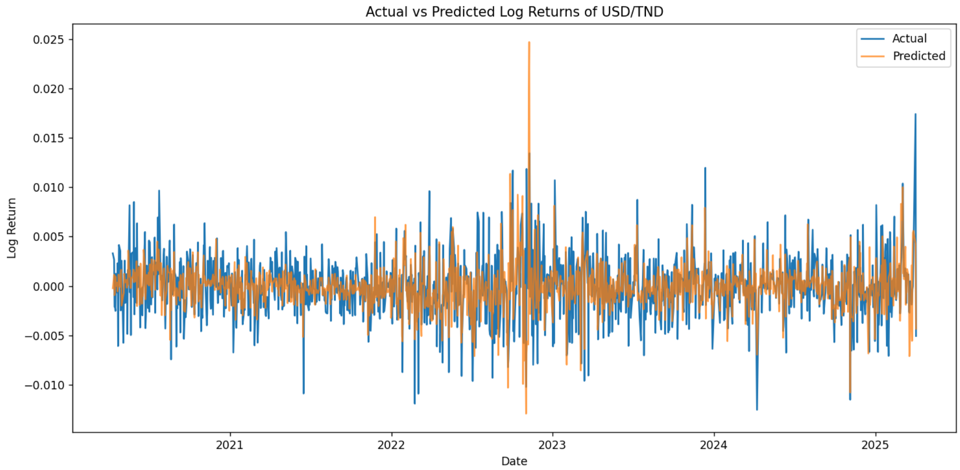


Figure 9: Actual vs Predicted Log Returns of USD/TND

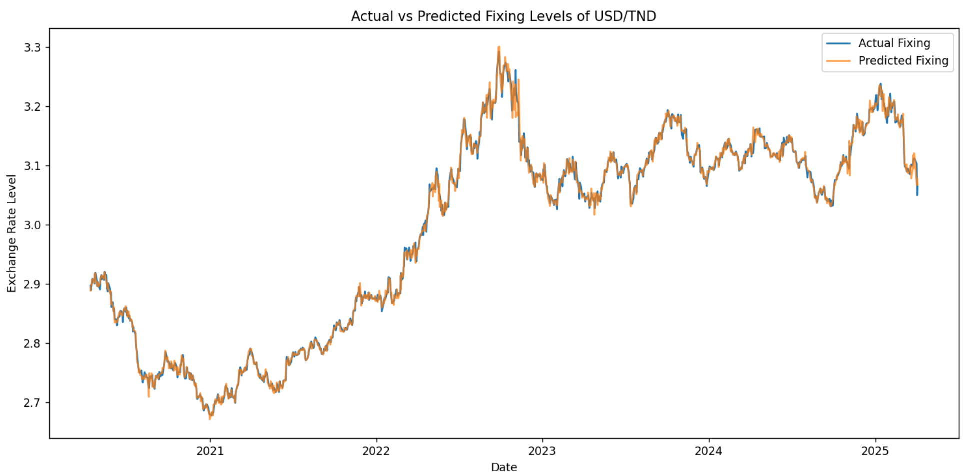


Figure 10: Actual vs Predicted Fixing Levels USD/TND

Here we notice that despite our use of `SGDRegressor`, the estimated weights remained constant throughout the entire period analyzed. The approximate constant weights found were -0.31 for EUR/USD, -0.05 for GBP/USD, and -0.01 for USD/JPY. Although the plots of actual versus predicted log returns and fixing levels visually appear to align well with the actual data, the fundamental issue found with this first approach is the failure to estimate time-varying weights. It is clear that a different methodology is required to capture the dynamic nature of the basket weights, as our hypothesis states.

6 Third Approach: Machine Learning Model with Time-Varying Weights

To address the limitation of our first approach, which failed to produce time-varying weights, our third approach focuses on using machine learning to predict these dynamic weights. The idea is to create a supervised machine learning dataset where the time-varying weights themselves are the target variable.

Our methodology comprised two main steps:

1. Creating a Dataset with the Target Weights

Instead of directly predicting the fixing's log return, the goal is to estimate the weights at each time step. To achieve this, we invert the daily fixing equation using OLS regression to determine what weights would have explained the actual price movement on that specific day.

The model uses the past 5 days of FX returns as input features and computes an approximate target vector of weights by inverting the fixing equation for each day using OLS.

This approach of using an economic model to generate targets for a machine learning model was inspired by the concept of *"target engineering"*[1].

The formula used for this inversion is based on simple returns:

$$\text{Fixing}_t = \text{Fixing}_{t-1} \cdot (1 + w_1 r_1 + w_2 r_2 + w_3 r_3)$$

where r_i represents the simple return of currency i.

OLS is used daily to compute the target weights that best fit the observed fixing change. The resulting weights exhibit variability over time when plotted, reflecting the dynamic nature we aim to capture.

[1] `https://bradleyboehmke.github.io/HOML/engineering.html`

2. Training a Nonlinear Machine Learning Model

A nonlinear model is then trained to learn the mapping from historical FX return features to the estimated daily weights obtained in the previous step. We used the `RandomForestRegressor` from the `scikit-learn` library.

The dataset is split into 80% training and 20% testing sets. The model is trained on the training data to predict the generated weights, and predictions are made on the test set. These predicted weights are then used to reconstruct the fixing levels using the previous day's actual fixing.

The results show that the model was able to predict the OLS-derived target weights smoothly. The RMSE for weight prediction was very low, at 0.001839, indicating a small prediction error. Furthermore, the predicted fixing levels, when reconstructed, visually followed the actual fixing levels closely.

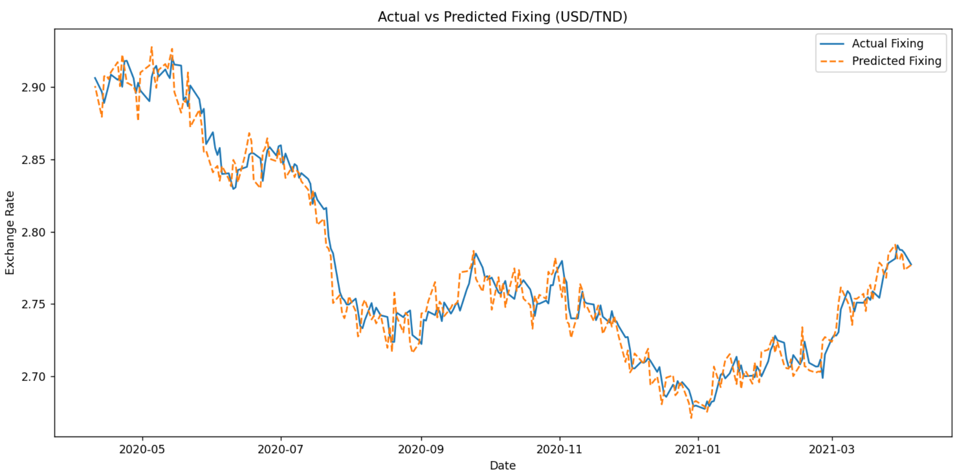


Figure 11: Actual vs Predicted Fixing

Figure 12: Estimated Weights Over Time (via TI)

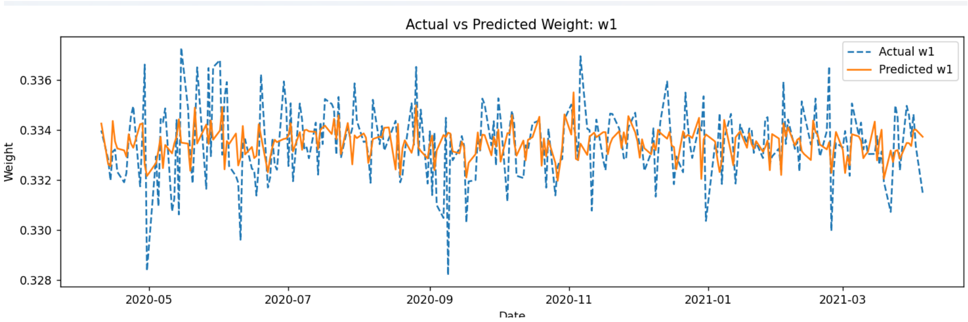


Figure 13: Actual vs Predicted w1

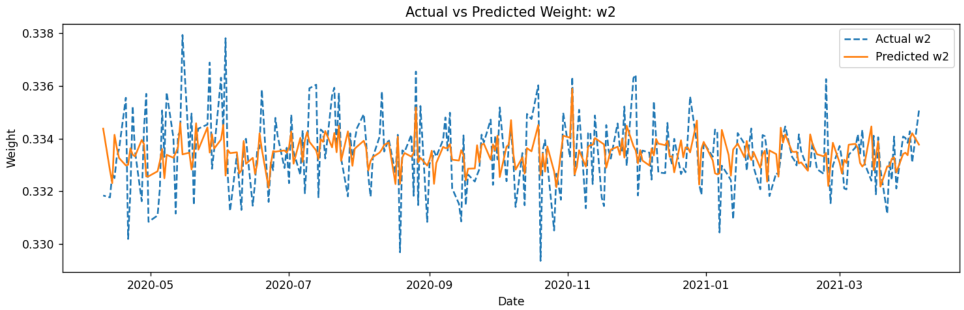


Figure 14: Actual vs Predicted w2

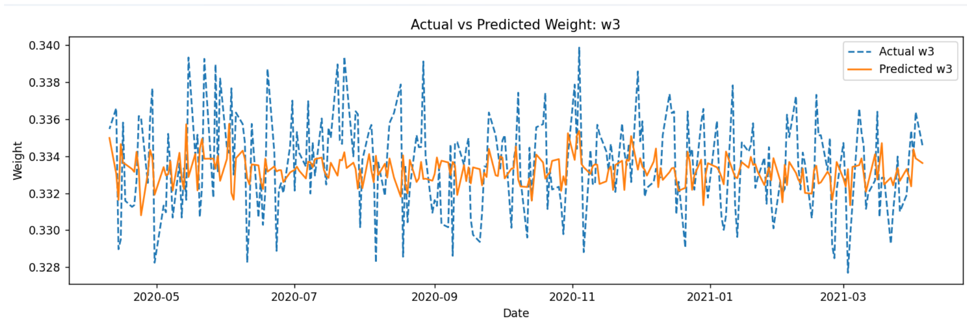


Figure 15: Actual vs Predicted w3

Despite the model's success in predicting the engineered target weights, the predicted weights tend to average around 30% each—indicating that while the model could reproduce the targets generated by the daily OLS inversion, those targets themselves may not represent the true dynamic weights required for optimal forecasting.

This outcome suggests that a different modeling framework or a more appropriate definition of dynamic weights is needed to achieve better accuracy than that of a simple constant-weight model.

7 Strategy Shift

After our findings with the three approaches, we introduced changes to our modeling approach driven by the recognition that a different methodology was necessary to achieve the desired accuracy improvements.

The core conceptual changes implemented focus on two main parts:

1. Accurately Calculating the Expected Fixing Today

Determining precisely the expected fixing value for the current day. Instead of predicting engineered OLS weights, the approach refines the formula for the Fixing calculation itself. Today's Fixing (*Fixing today*) should be based on yesterday's value (Fixing_{t-1}) multiplied by a component involving time-varying, stochastic weights ($w_{i,t}$) applied to FX log returns $\log(R_t) - \log(R_{t-1})$. The formula includes associated error terms ($\epsilon_{i,t}$ and $E_{i,t}$):

$$\text{Fixing}_t = \text{Fixing}_{t-1} \cdot \left(\sum_i w_{i,t} \cdot \log\left(\frac{R_{i,t}}{R_{i,t-1}}\right) + \epsilon_{i,t} \right) + E_{i,t}$$

A crucial strategic change is the proposal to build a nonlinear machine learning model that is stochastic, in order to directly predict these time-varying weights and errors. Exploring a Kalman Filter is a potential methodology for modeling these dynamic and stochastic elements. The goal remains to minimize the error percentage and demonstrate superiority by comparing the results to OLS.

2. Addressing the Difference Between Predicted Fixing and Market Rate

Here we acknowledge a known issue: the calculated Fixing value often does not perfectly match the actual market rate or interbank rate.

To address the spread, we lean into using nonlinear machine learning stochastic methods. This aims to account for factors like spikes and illiquidity that contribute to this divergence.

Our objective is to minimize the error percentage associated with forecasting this spread as a separate component. We move away from predicting retrospectively calculated static or static-like targets towards directly modeling stochastic components that contribute to the final exchange rate.

8 Kalman Filter Approach

The initial multiplicative model relating the current day's fixing to the previous day's fixing and weighted returns of a currency basket is transformed into a linear state-space model. This is achieved by focusing on the return ratio, defined as:

$$\text{Fixing}_t = \text{Fixing}_{t-1} \cdot (w_{1,t} \cdot \log(\text{EUR}_t) + w_{2,t} \cdot \log(\text{GBP}_t) + w_{3,t} \cdot \log(\text{JPY}_t))$$

This ratio is then modeled as a linear combination of the log levels of the basket currencies (EUR, GBP, JPY) plus noise:

$$r_t = x_t^T w_t + \text{noise}$$

The model is formally defined with two main equations:

- **Observation Equation:**

  $$r_t = X_t w_t + v_t$$

  where r_t is the observed return ratio, X_t is the matrix containing the log levels of EUR, GBP, and JPY at time t, w_t is the vector of time-varying weights, and v_t is the observation noise.

- **State Equation:**

  $$w_t = w_{t-1} + u_t$$

  which assumes the time-varying weights w_t follow a random walk process, where u_t is the process noise.

Implementation

The Kalman Filter is used to estimate the time-varying weights. It calculates the minimum-mean-square-error (MMSE) estimates of the state (weights) based on the observed data up to the current time step.

The implementation involves:

- Defining the transition matrix (identity matrix, assuming weights do not change systematically),
- Observation matrices (the matrix of log FX levels),
- Initial state mean (zero),

- Initial state covariance and transition noise covariance matrices.

Training

The filter's built-in Expectation-Maximization (EM) algorithm is utilized to automatically fit the process noise covariance matrix Q and the observation noise covariance matrix R by maximizing the likelihood of the observed data. The code executes the EM algorithm for 10 iterations.

Filtering and Forecasting

Afterwards, the data is processed chronologically, producing online estimates of the weights and their associated covariances. These filtered weight estimates are used to forecast the return ratio via `kf.filter`. The predicted fixing level is subsequently calculated by anchoring it to the actual fixing from the previous day.

Diagnostics and Results

The performance of the model is evaluated using standard metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Visual diagnostics support the model's validity.

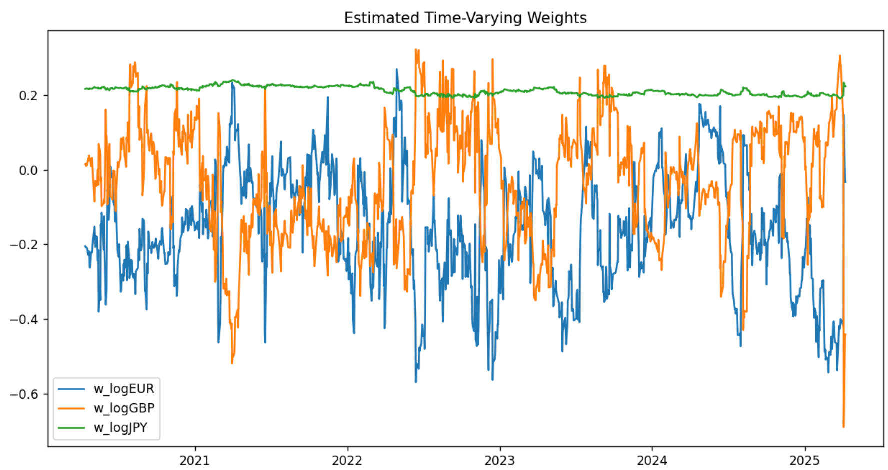


Figure 16: Estimated time-varying weights over time, demonstrating fluctuation and supporting the premise that constant weights are suboptimal.

Version 2: Log-Return Transformation

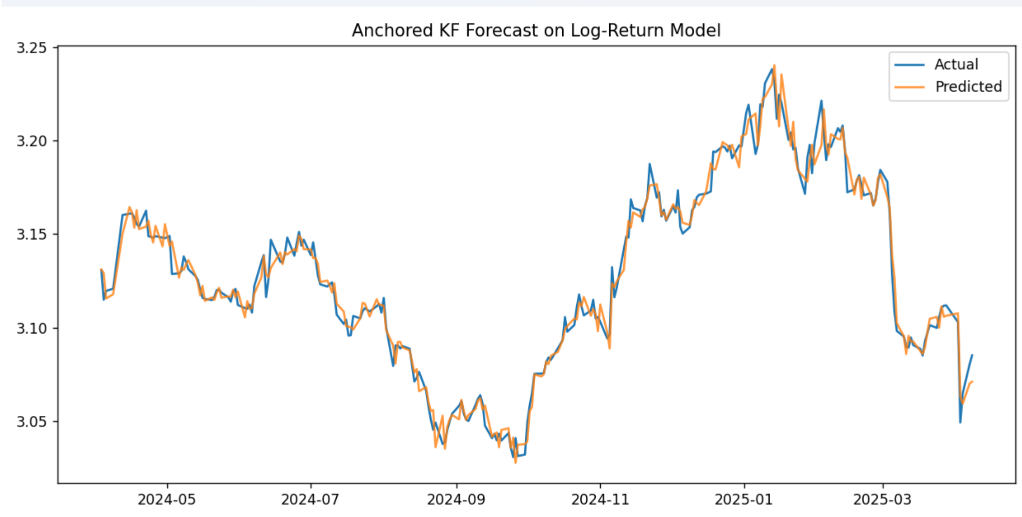
Afterwards, we began the transformation of both the features and the target to log-returns. Instead of using the log levels of EUR, GBP, and JPY as features and the return ratio as the target, we use the daily log-returns as features and the daily log-return of the Fixing as the target.

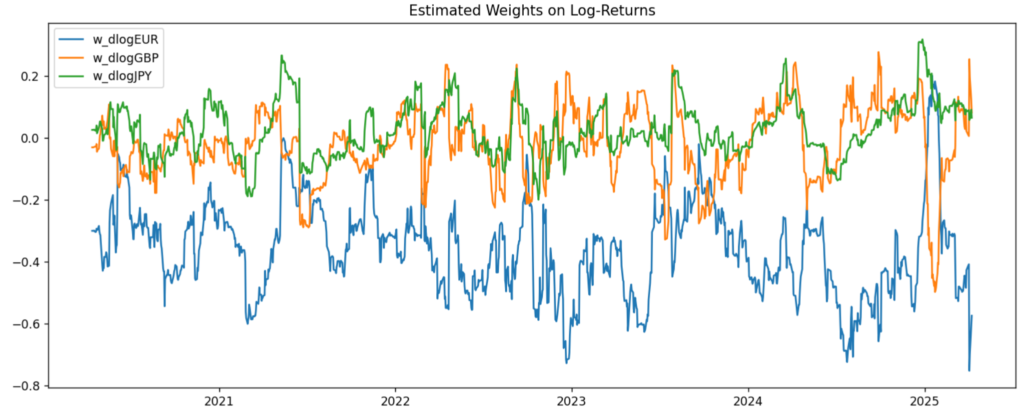
This means our observation equation for the Kalman Filter changes from being based on the return ratio and log levels to being based on log-returns:

$$\Delta \log(\text{Fixing}_t) = w_t^T \cdot \Delta \log(X_t) + v_t$$

Results:

- Out-of-Sample RMSE: 0.006833
- Out-of-Sample MAPE: 0.166%





```
Last 5 weight estimates on log-returns:
             w_dlogEUR  w_dlogGBP  w_dlogJPY
Date
2025-04-02  -0.408129   0.005867   0.089335
2025-04-03  -0.751896   0.254112   0.057761
2025-04-04  -0.699652   0.196293   0.060834
2025-04-07  -0.602008   0.120964   0.091006
2025-04-08  -0.574200   0.081895   0.065052
```

Version 3: Manual Grid Search for Q and R

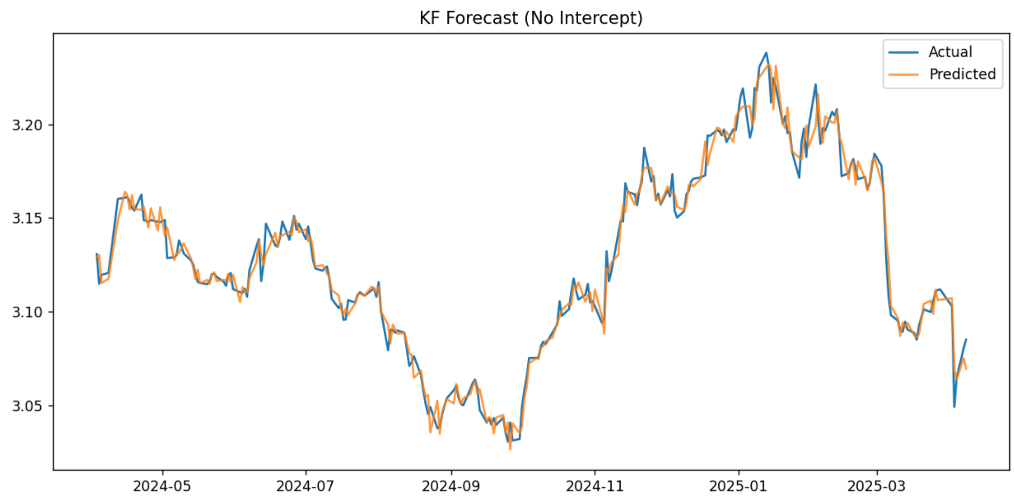
The main innovation here is replacing the Kalman Filter's built-in Expectation-Maximization training for the covariance matrices Q (process noise) and R (observation noise) with a manual grid search.

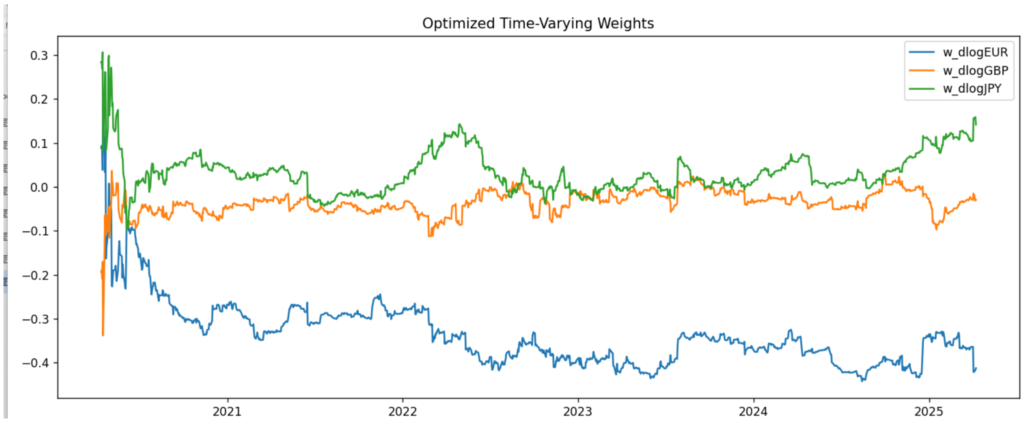
The script defines grids of potential values for Q and R. It then iterates through all combinations, runs the Kalman Filter, and calculates the out-of-sample performance (RMSE, MAPE) for each pair of (Q, R).

The pair of (Q, R) that yields the best out-of-sample MAPE is selected as the optimal setting. It also explicitly notes the model is run with *no intercept*.

Results:

- Best Hyperparameters Found: $Q = 5 \times 10^{-6}$, $R = 1 \times 10^{-6}$
- RMSE: 0.007273
- MAPE: 0.169%





Version 4: GARCH-Based Time-Varying Observation Noise

We then introduced a time-varying observation noise covariance R_t. This is achieved by first fitting an OLS model on the training log-returns to obtain residuals. Then, a GARCH(1,1) model is fitted to these OLS residuals to model their conditional volatility, which is used as the time-varying R_t.

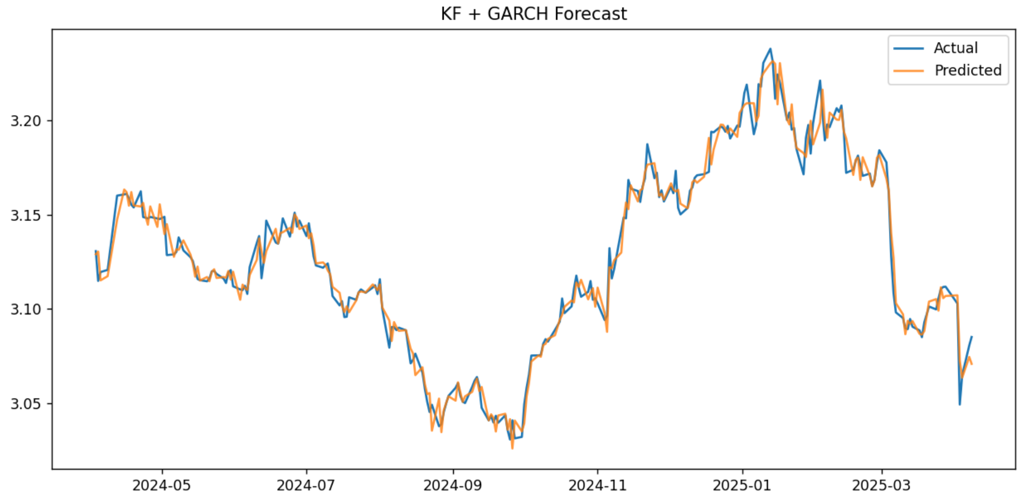
The Kalman Filter implementation here is a manual loop, which allows it to incorporate the daily R_t value from the GARCH model. Because R_t is now determined by GARCH, the grid search is only performed over the process noise covariance Q.

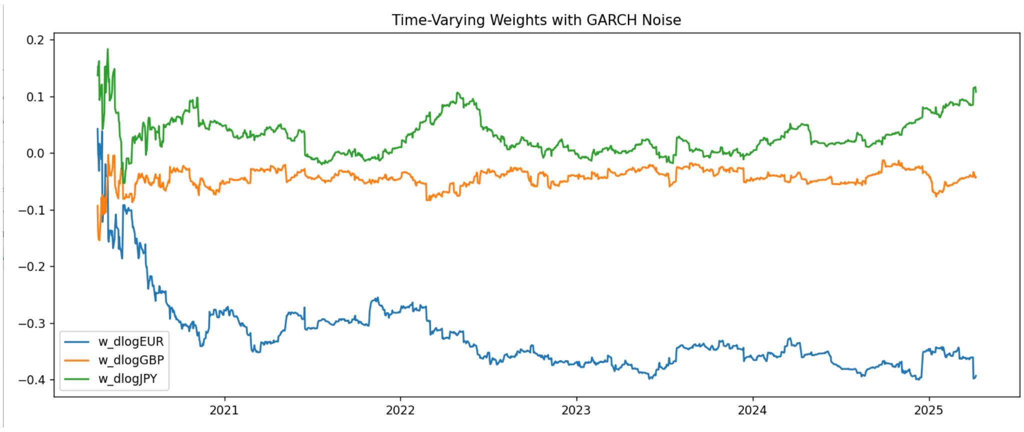
This approach aims to make the filter more responsive to periods of higher volatility in the observation errors by letting R_t fluctuate according to the GARCH model.

Results:

- Best Found: $Q = 1 \times 10^{-5}$
- RMSE: 0.007035
- MAPE: 0.165%

Together these confirm that the time-varying weights are stable and interpretable, and the GARCH-weighted Kalman Filter one-step forecast is extremely tight. Visually and quantitatively, this model is performing very well.






```
Last 5 weights:
            w_dlogEUR  w_dlogGBP  w_dlogJPY
Date
2025-04-02  -0.361037  -0.042453   0.085000
2025-04-03  -0.398611  -0.033446   0.114780
2025-04-04  -0.398545  -0.035431   0.114955
2025-04-07  -0.396185  -0.043986   0.116519
2025-04-08  -0.393526  -0.042868   0.107829
```

9 Double Kalman Filter: Joint Estimation of Basket Weights and Spread

Building upon the Kalman Filter framework for estimating time-varying basket weights, we adopted a more sophisticated approach to directly model both the underlying currency basket dynamics and the spread between the USD/TND Fixing rate and the Interbank rate within a single filter. This method is referred to as the *Double Kalman Filter*, and we selected it for its elegant, all-in-one mechanism that captures both the stochastic evolution of the basket weights and a latent liquidity-spread state.

State-Space Formulation

The core idea is to extend the hidden state vector θ_t beyond just the basket weights. The latent state adapted to the information set includes both the time-varying weights for the major currency pairs and a hidden spread component.

The model is defined by the following discrete-time state-space equations:

- **Transition Equation:**

$$\theta_t = \theta_{t-1} + q_t, \quad q_t \sim \mathcal{N}(0, Q)$$

  This assumes that both the weights and the spread state follow a random walk, meaning their values at time t are equal to their values at $t-1$ plus some noise. The process noise covariance matrix Q is diagonal, and the spread state is often allowed a larger variance to better track market volatility.

- **Observation Equation:**

$$y_t = \Delta \log(F_t) = \log\left(\frac{F_t}{F_{t-1}}\right) = w_t^\top X_t + s_t + v_t$$

  where w_t is the time-varying weight vector, X_t is the FX log return vector, s_t is the latent spread state, and $v_t \sim \mathcal{N}(0, R)$ is observation noise.

Kalman Filter Recursion and Estimation

The Kalman Filter is applied to this state-space model. It recursively computes the minimum-mean-square-error (MMSE) estimates of the hidden state vector θ_t given all available observations up to time t. This yields filtered estimates for both the basket weights and the latent spread state across time.

Forecasting the Fixing and Interbank Rates

The filtered state estimates are used to generate forecasts:

- The predicted log return of the Fixing rate at time t is computed using the filtered weights and spread.

- The predicted Fixing level is then reconstructed by anchoring it to the previous day's actual Fixing rate.
- The predicted Interbank rate is obtained by adding the estimated latent spread state s_t to the predicted Fixing rate.

Hyperparameter Tuning

The performance of the Kalman Filter strongly depends on specifying appropriate values for the process noise covariance Q and the observation noise variance R. Instead of relying solely on the Expectation-Maximization (EM) algorithm, we performed a manual grid search across plausible values of Q and R.

The objective was to minimize the out-of-sample Mean Absolute Percentage Error (MAPE) for the Interbank rate forecast. Through this grid search, optimal values for Q and R were identified.

Results

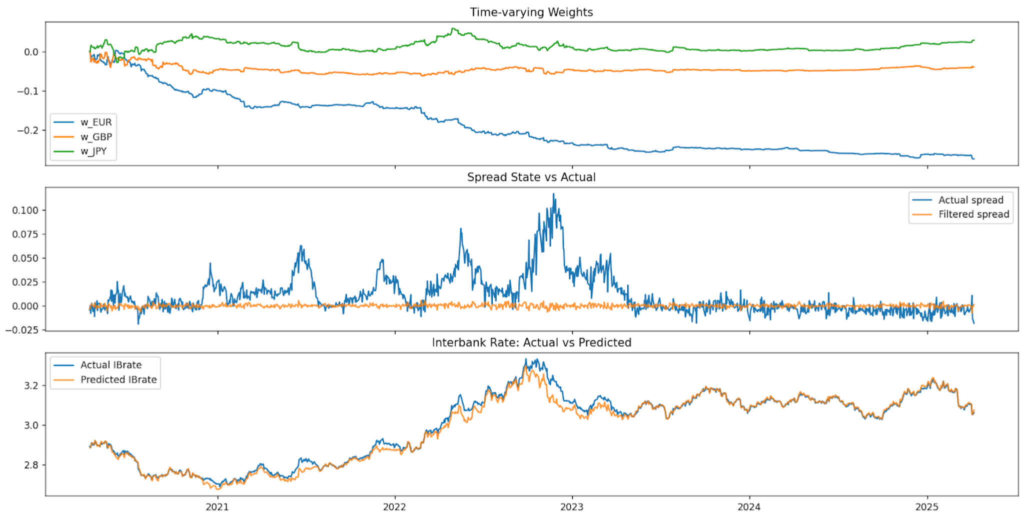
The Double Kalman Filter approach resulted in significant performance improvements over prior methods. Evaluating out-of-sample forecasts on the test set yielded the following metrics:

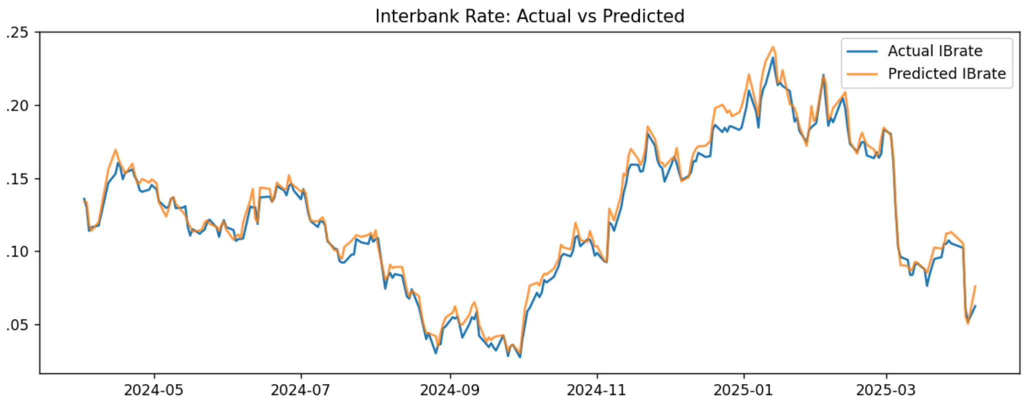
Fixing Forecast:

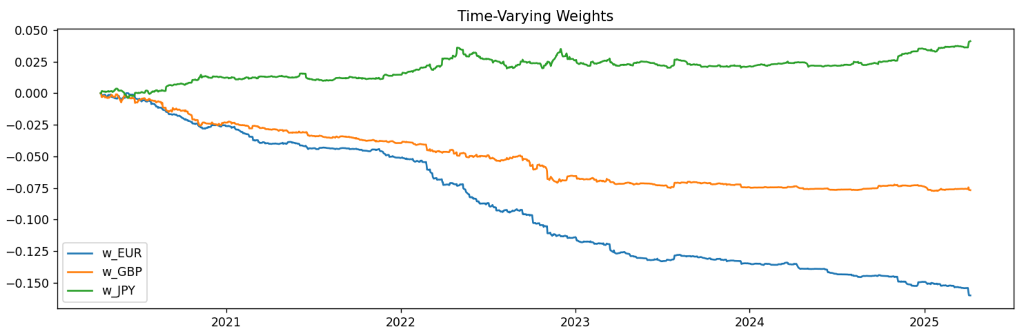
- RMSE: 0.012046
- MAPE: 0.286%
- Correlation: 0.9711
- Directional Accuracy: 48.61%

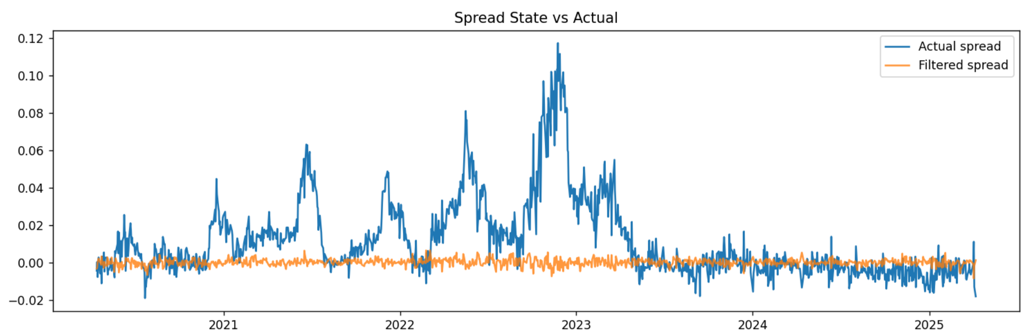
Interbank Forecast:

- RMSE: 0.01366
- MAPE: 0.333%
- Correlation: 0.9678
- Directional Accuracy: 58.17%


Time-varying Weights
w_EUR
w_GBP
w_JPY
Spread State vs Actual
Actual spread
Filtered spread
Interbank Rate: Actual vs Predicted
Actual IBrate
Predicted IBrate
2021
2022
2023
2024
2025


Interbank Rate: Actual vs Predicted
Actual IBrate
Predicted IBrate
.25
.20
.15
.10
.05
2024-05
2024-07
2024-09
2024-11
2025-01
2025-03


Time-Varying Weights
w_EUR
w_GBP
w_JPY
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−0.150
2021
2022
2023
2024
2025


Spread State vs Actual
Actual spread
Filtered spread
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
2021
2022
2023
2024
2025

The significant reduction in MAPE for the Interbank rate prediction demonstrated the effectiveness of jointly modeling the components within the Kalman framework.

The high correlation (above 0.9711) for both Fixing and Interbank forecasts, along with directional accuracy up to 58.17% for Interbank (compared to 53.8% previously), indicates that this model provides a higher accurate and coherent baseline for simultaneous prediction.

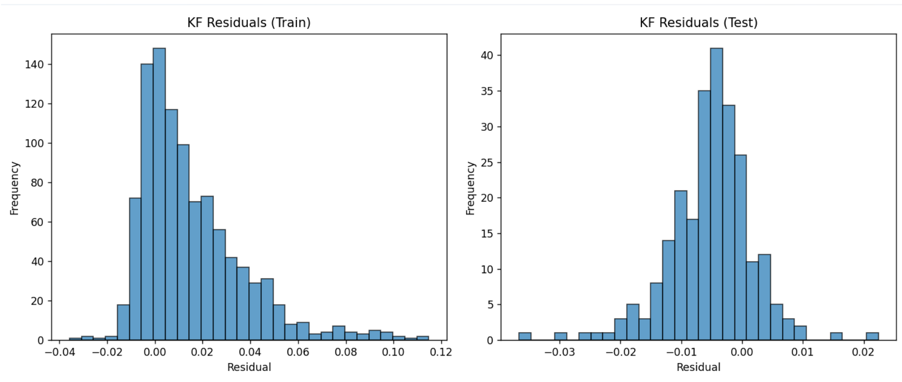
The time-varying weights estimated by the filter remained stable and interpretable.

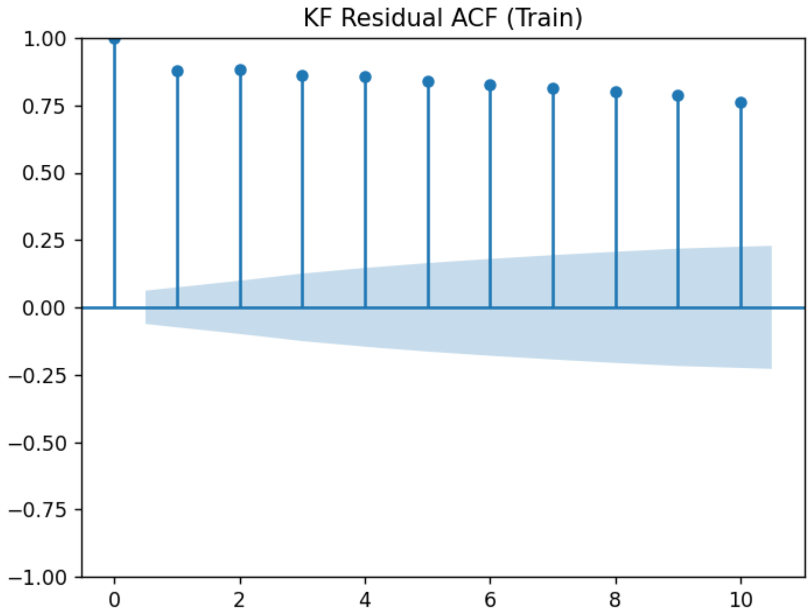
This Double Kalman Filter model successfully provides a sophisticated and robust baseline prediction for both the USD/TND Fixing and Interbank rates by simultaneously estimating dynamic basket weights and the underlying liquidity spread state.

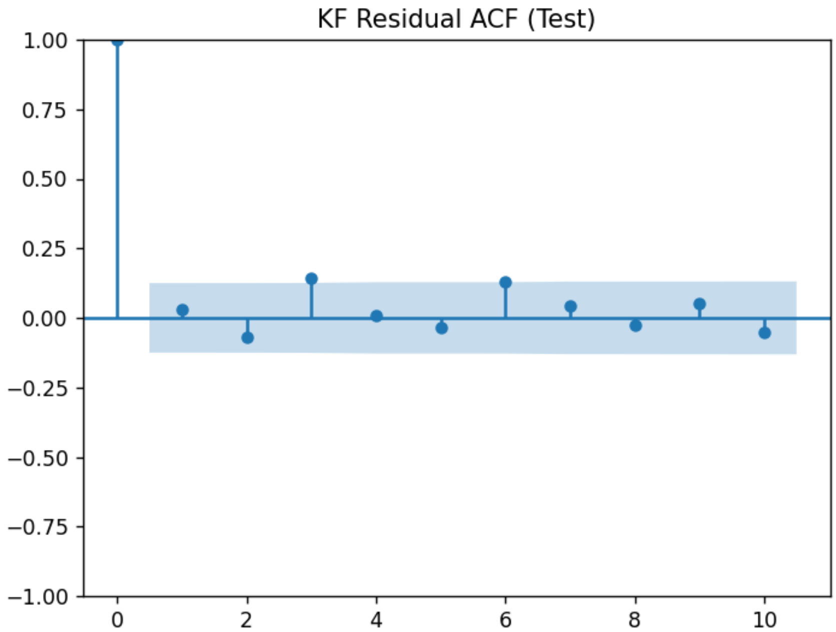
This effectively concludes the basket-based modeling phase by producing a highly accurate forecast that inherently accounts for the spread component, setting the stage for potential further refinement using residual modeling.

9.1 Goodness of Fit

Since we know that the weights in the short run vary very slowly and are nearly constant, it is normal to find that the Kalman Filter is underfitted. As noted, the weights are nearly, but not totally, constant, thus, we are using a very small process noise parameter $Q = 10^{-5}$. The goal is to let the Kalman Filter only capture the necessary long-term movement and prevent it from chasing unnecessary noise. At the same time, the Neural Network is expected to handle the remaining variation in the short run. This modeling rationale justifies our decision to use a hybrid approach.








```
=== KF Baseline Performance ===
In-sample     → RMSE: 0.02669, MAPE: 0.5999%
Out-of-sample→ RMSE: 0.00855, MAPE: 0.2084%

=== Hybrid (KF + NN) Performance (Out-of-sample) ===
RMSE: 0.00651, MAPE: 0.1514%
```

Observations

Even though the ACF of the training set shows high autocorrelation and the residuals are left-skewed—both signs of underfitting—this behavior is entirely consistent with how our model is structured.

In essence, these diagnostics reveal that the Kalman Filter excels at capturing short-run variations in the weights. However, it struggles with long-term dynamics, as the training data spans multiple regimes, cycles, and economic periods. Over such extended periods, the model cannot adapt quickly due to its intentionally small Q value.

Conversely, when evaluated on a shorter testing window, the model shows no auto-correlation or skewness in residuals, indicating a good fit. This aligns with the core goal of our framework: to nowcast the interbank rate using dynamic weight adjustment on short horizons.

- **Long-period weight estimation:** Underfitted model
- **Short-period weight estimation:** Well-fitted model

10 Performance Evaluation & Backtesting

After the different approaches were explored, our objective is to quantify the accuracy and robustness of the models in out-of-sample nowcasting of the USD/TND Fixing and, critically, the Interbank rate, which reflects the real-time market conditions.

Methodology

Performance evaluation was conducted using a standard time-series split. The available data was divided into an 80% training set and the subsequent 20% test set. Models were trained exclusively on the training data and then evaluated on their ability to forecast the test data, simulating a real-world forecasting scenario where predictions are made on unseen future data.

For evaluating the forecasts, especially for the Fixing rate derived from the basket model and the baseline Interbank forecast from the Kalman Filter, an anchored one-step-ahead prediction approach was used. This means that the prediction for the current day's level was computed by applying the model's predicted log-return (or predicted return ratio) to the actual observed level from the previous day.

Key performance metrics were employed to provide a comprehensive view of model accuracy:

- **Root Mean Squared Error (RMSE):** Measures the square root of the average of squared differences between predicted and actual values, providing a measure of the typical error magnitude.
- **Mean Absolute Percentage Error (MAPE):** Represents the average absolute percentage difference between predicted and actual values, offering a clear metric relative to the scale of the series. This was a primary metric used for model tuning and comparison.
- **Correlation ρ:** Indicates the linear relationship between the actual and predicted time series, showing how well the forecast tracks the overall movement.
- **Directional Accuracy:** The percentage of days where the predicted direction of change (up or down) matches the actual direction of change.
- **Mean Absolute Error (MAE):** Similar to RMSE but measures the average absolute difference, often used alongside RMSE.

Evolution of Model Performance

Our work involved exploring several modeling strategies to forecast the Fixing and Interbank rates, progressively refining the approach based on performance evaluations.

Constant OLS Regression: A basic Ordinary Least Squares regression using log returns served as an initial benchmark for predicting the Fixing rate with constant weights. This model yielded an Out-of-Sample RMSE of 0.009795 and MAPE of 0.217%. While simple, its performance was described as ”respectable”.

Machine Learning for Dynamic Weights — Random Forest: Initial attempts to predict dynamic weights using Machine Learning models like Random Forest, trained on engineered targets derived from daily OLS inversion, showed promisingly low RMSE for weight prediction around 0.001839. However, the resulting weights were clustered around average values, and reconstructing the Fixing forecast from these weights did not yield significant improvements. This approach was eventually superseded by the Kalman Filter for weight estimation.

Kalman Filter for Time-Varying Weights: Transitioning to a Kalman Filter approach for estimating time-varying weights showed significant improvements in forecasting the Fixing rate.

- Refining the model to use log-returns as both features and the target for the Kalman Filter resulted in an Out-of-Sample RMSE of 0.006833 and MAPE of 0.166% for the Fixing forecast.

- Optimizing the Kalman Filter's process and observation noise covariances Q and R via grid search found optimal values ($Q = 5 \times 10^{-6}, R = 1 \times 10^{-6}$) yielding Fixing performance of RMSE = 0.007273 and MAPE = 0.169%. Even though this creates higher error rates, it only focuses on crucial information rather than noise. We later use a final version that refines this model.

- Incorporating a GARCH(1,1) model to dynamically estimate observation noise further refined the Fixing forecast, achieving optimal metrics of RMSE = 0.007035 and MAPE = 0.165%. This KF+GARCH model visually and quantitatively performed very well on the Fixing.

Spread Nowcasting — ARIMAX vs Random Forest: To forecast the Interbank rate, the spread between the Fixing and Interbank was analyzed. Initial analysis showed the spread time series to be stationary, suggesting suitability for time series models.

- An ARIMAX(1,0,1) model incorporating FX log returns improved performance over a pure AR(1), yielding a Spread Nowcast RMSE of 0.015854 and MAE of 0.014649. However, its forecast still exhibited bias and smoothed over volatility spikes.

- Switching to a Random Forest Regressor for spread nowcasting, using features like lagged spread, lagged Fixing/IBrate, and spread volatility, provided a **"HUGE PERFORMANCE BOOST"**. This model achieved a Spread Nowcast RMSE of 0.006499 and MAE of 0.004896. Feature importance analysis confirmed that the lagged spread (`spread_lag1`) was the most important feature, explaining 96.7% of the variance. Train vs Test error comparison showed no overfitting.

Combined KF + GARCH + RF: The initial combined model, using KF+GARCH for Fixing and RF for Spread, provided the first full Interbank rate forecast. This yielded a Fixing forecast RMSE of 0.006993 (MAPE 0.165%) and an Interbank forecast RMSE of 0.007863 (MAPE 0.190%). It showed high correlation (around 0.9893) but limited directional accuracy (52.6%).

Double Kalman Filter: A significant step was the implementation of a Double Kalman Filter that jointly estimates both the basket weights and a latent spread state. This elegant all-in-one approach resulted in a huge increase in performance.

- Initial results showed Fixing RMSE 0.012 (MAPE 0.286%) and Interbank RMSE 0.0136 (MAPE 0.333%). Both series showed very high correlations (0.9711 for Fixing, 0.9678 for Interbank) and substantially good directional accuracies (48.61% for Fixing, 58.17% for Interbank).

- Grid search for optimal Q and R parameters in the Double Kalman filter confirmed similar performance, finding optimal $Q = 1 \times 10^{-5}, R = 0.001$.

Regime-Aware Spread Nowcast: An enhancement to the combined model involved training separate Random Forest models for spread nowcasting based on detected market regimes (normal vs. stressed, using lagged spread volatility). This marginally improved the Interbank directional accuracy (from 52.6% to 53.8%) but did not significantly alter overall error metrics (MAPE slightly increased to 0.1910%).

Final Model Performance: Weights and Interbank Nowcasting Mechanism

The final, best-performing model combines the Double Kalman Filter's capabilities with a Neural Network for residual correction. The Double Kalman Filter provides a baseline forecast for the Interbank rate by combining the Fixing forecast (derived from estimated weights and FX log returns) and the filtered spread state (s_t).

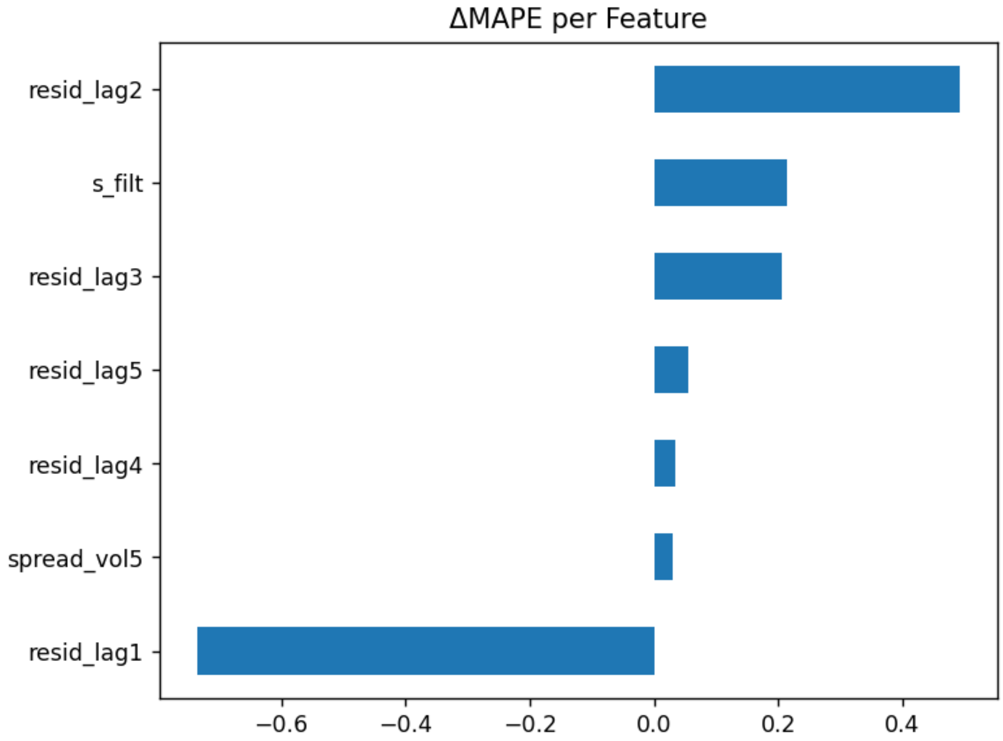
A Multilayer Perceptron (MLP) Neural Network is then trained to model and predict the residuals between the actual Interbank rate and this KF baseline forecast. The NN uses features such as lags of the residuals, realized spread volatility, and the filtered spread state from the Kalman filter. The final hybrid forecast is the sum of the KF baseline forecast and the NN's predicted residual.

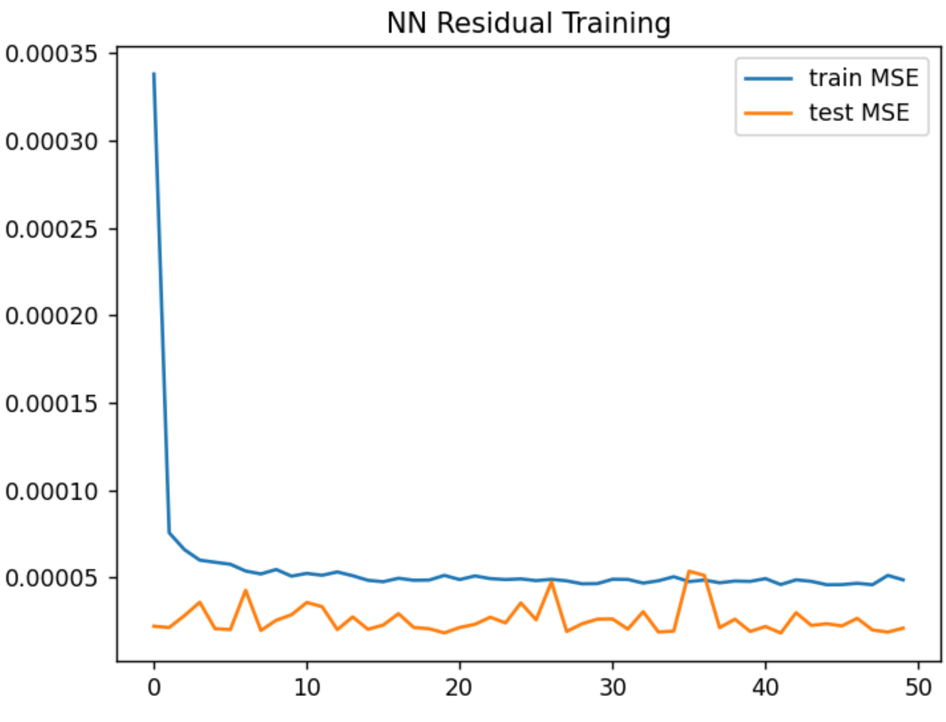
Series	**RMSE**	**MAPE (%)**	**Correlation**	**DirAcc (%)**
Fixing	0.0081	0.1879	0.9867	55.102
Interbank	0.0074	0.1761	0.9912	65.7143
Hybrid	0.0063	0.1452	0.9919	66.9388

Table 3: Final Model Performance Summary

The Neural Network residual correction successfully reduced the Interbank MAPE from 0.333% (pure Kalman baseline) to 0.1452% in the hybrid model. The RMSE also decreased from 0.013 to 0.0063. The hybrid model maintains a very high correlation of 0.9919 with the actual Interbank rate, and a higher Directional accuracy of 66.938%

The NN training process showed decreasing training and test MSE over epochs, indicating learning and a lack of significant overfitting on the residuals. Feature importance analysis for the NN revealed that lagged residuals (`resid_lag1`, `resid_lag2`, `resid_lag3`) and the filtered spread state (s_{filt}) were the most influential predictors of the residual.

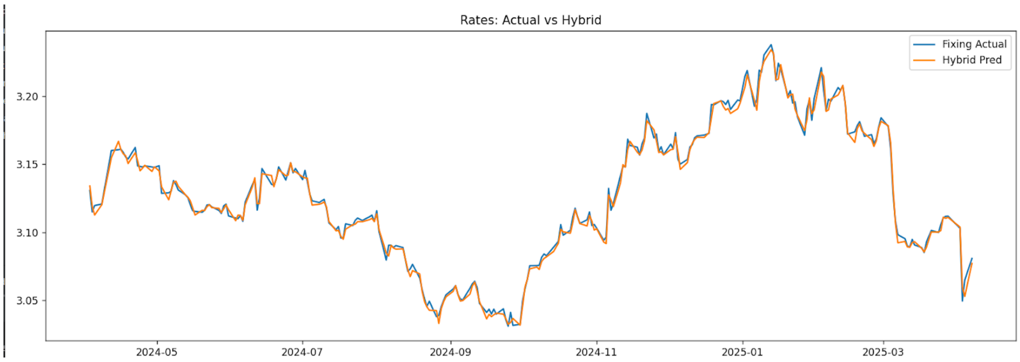




Visual Diagnostics: Stochastic Adjustment Integration

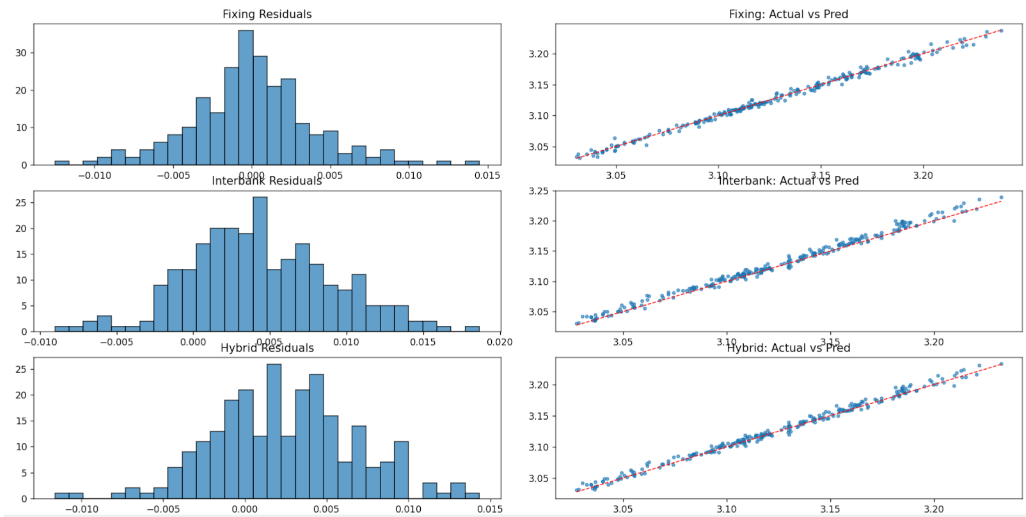
Various plots were used to visually assess the model's performance:

Actual vs. Predicted Time Series: Overlays of the actual Interbank rate and the final Hybrid forecast show a close visual fit, indicating that the model effectively tracks the market rate over time. Similar plots for the Fixing forecast demonstrate the KF's accuracy on the basket component.

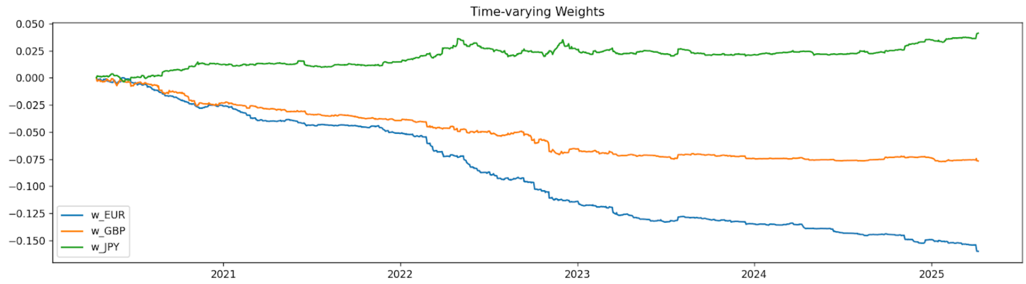


Residual Histograms: Histograms of the prediction errors for both Fixing and Interbank/Hybrid forecasts are centered around zero, as expected for a well-calibrated model. However, they show heavier tails, particularly during volatile periods, indicating larger errors occur during market stress.

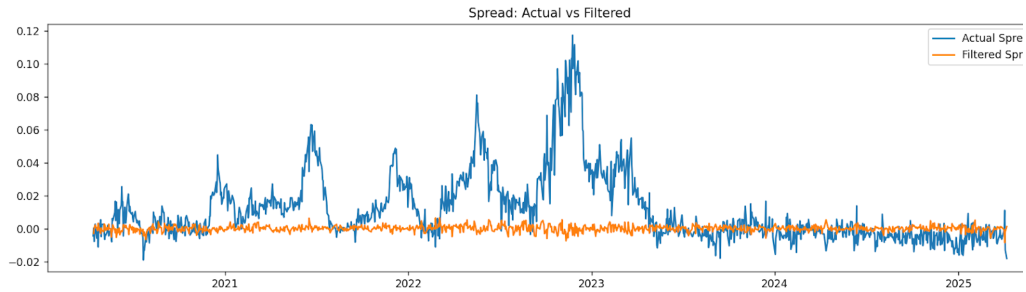
Actual vs. Predicted Scatterplots: Scatterplots confirm the high correlation, with predicted values clustering tightly around the 45-degree line representing perfect prediction.



Time-Varying Weights: Plots of the estimated weights from the Double Kalman Filter demonstrate their dynamic evolution over time, supporting the initial hypothesis that weights are stochastic.



Spread (Actual vs. Filtered/Predicted): Plots comparing the actual spread to the filtered spread state from the Double Kalman Filter or the predicted spread from the RF/Regime-aware RF show how well the model components capture the dynamics of the spread, including its spikes. The RF spread nowcast, while significantly better than ARIMAX, still struggles to perfectly capture the largest spikes.



Discussion

The rigorous backtesting process demonstrates the effectiveness of the developed models. The Double Kalman Filter successfully estimates dynamic basket weights and a latent spread state, providing a strong baseline forecast for both the Fixing and Interbank rates. The introduction of the Neural Network as a residual correction mechanism further significantly improves the Interbank rate nowcasting accuracy, notably reducing the Mean Absolute Percentage Error.

While the correlation and magnitude of error (RMSE, MAPE) are excellent, directional accuracy, though improved by the NN, still presents a challenge, being only moderately above a coin flip. This suggests that predicting the precise sign of daily price changes, especially in the volatile spread component, remains difficult. The analysis of residuals and spread plots indicates that the largest errors occur during periods of significant local market shocks or illiquidity, which manifest as large spread spikes that are hard for the models to fully anticipate or correct.

In conclusion, the Kalman Filter and Neural Network hybrid model provides a robust and accurate framework for nowcasting the USD/TND interbank rate in real-time. The layered approach, combining a fundamental basket model with state-space dynamics and a data-driven residual correction, successfully addresses the complexities of FX valuation in a market influenced by both global and local factors.

Limitations

It is important to acknowledge that the accuracy of the model is inherently constrained by the limitations of the available data. Specifically, the IBrates, FixingRates, and currency basket data used in this study are halted at April 7, 2025. This temporal cutoff restricts the model's ability to reflect the current market dynamics and limits its reliability in a real-time forecasting context.

For the model to achieve its full potential, the underlying database must be updated regularly with the most recent information. With an up-to-date and complete dataset, the framework can effectively extract real-time foreign exchange signals and nowcast the interbank rate with high precision.

General Conclusion

This study set out to build a real-time intrinsic valuation model for the Tunisian Dinar (TND) that could adapt to both global currency movements and local market frictions. What began as an exploration of traditional regression and machine learning approaches evolved into a deeper inquiry into dynamic, state-space modeling. Along the way, we challenged the assumption that exchange rate relationships are static, and instead treated the model's core components—the basket weights and the spread—as stochastic processes.

The turning point in our development was the implementation of the Double Kalman Filter. By jointly estimating time-varying weights and a latent liquidity spread, the model captured both global FX fundamentals and the local pressures often invisible in simpler models. Its performance speaks for itself, with out-of-sample results showing remarkably low forecasting errors and correlations exceeding 0.9918 with the actual market rates. It delivered not just accuracy, but consistency and interpretability.

To push the boundaries even further, we integrated a Neural Network that learned from the residual errors of the Kalman-based forecasts. This hybrid model reduced forecasting errors even more, especially for the interbank rate, and added a layer of flexibility that complemented the structure of the Kalman filter. While perfect directional accuracy remains elusive—particularly during periods of volatility and spread spikes—the system performed significantly better than chance, proving its utility in real-world scenarios.

Ultimately, the model we developed offers more than just numbers. It offers insight into the behavior of the TND in a market shaped by both external currencies and internal liquidity dynamics. And while there is always room for refinement—whether through regime-aware filters or volatility-adjusted forecasts—this work lays a solid foundation for real-time FX modeling in emerging markets, combining economic theory with statistical rigor and machine learning adaptability.