ML failures in financial time series

Evidence from USD 3M Rates and USD FX index forecasting

Youssef Louraoui

Université Paris-Saclay
20230348@etud.univ-evry.fr

More complex vs More accurate

Model	**Complexity**	**MAE score**
Random Forest	High	**1.215**
GBoost	High	2.156
SVM	Medium	3.193
MLP	Medium-High	3.443
LSTM	Medium-High	7.027

Key finding: Simple ensemble methods outperform neural networks by 66-83% on USD 3M rates

This challenges the conventional wisdom that neural networks are optimal for financial time series

Why we built this

Can ML predict interest rates and currency levels?

Industry belief

Deep learning $\rightarrow$ state-of-art
Neural networks capture nonlinearity automatically

Our finding

Simple models $>$ neural nets
Fed policy shifts destroy all predictions equally

Implication: ML models fail catastrophically during regime shifts (Fed tightening, pandemic shocks, etc.)

Experimental setup

Dataset

2,608 daily observations (2018-2024)
Targets: USD 3M rates (USD3MD=) & USD FX index (=USD)

Training: 1,825 observations (70%)
Testing: 783 observations (30%)
Normalization: Z-score (+/- 3 σ threshold)

Features: 20 daily FX + rates
- GBP, EUR, JPY, CHF, AUD, NZD, CAD, SEK, NOK
- 3M deposit rates for each
- **First 5 variables explain 95% of variance of targets (PCA)**

Challenge

Train on 2014-2020 (pre-pandemic and early outbreak) $\rightarrow$ Test end 2020-2024 (Fed tightening = regime shift)

Models tested

Neural Networks

- MLP: 3-layer perceptron (dropout 25%, epoch 10)

- LSTM: Recurrent network (sequence length 30)

Tree-Based + SVM

- Random Forest: 50 trees (depth 20)

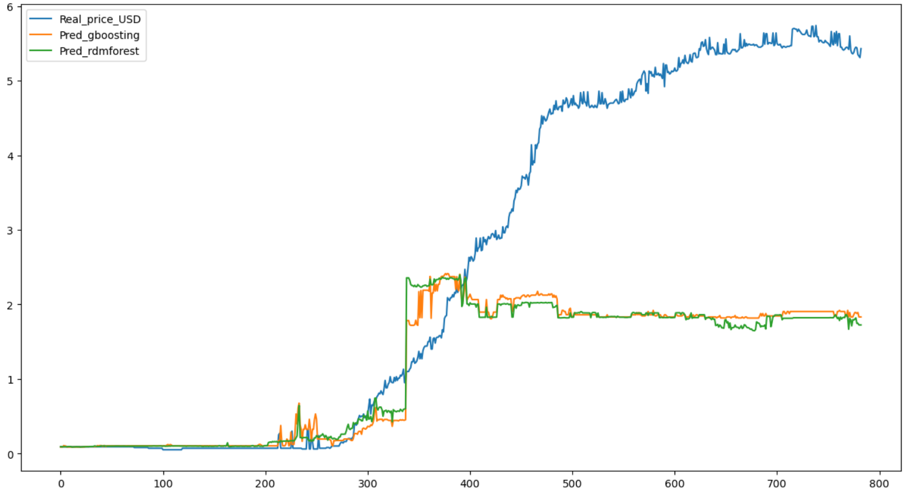
- Gradient Boosting: Sequential (learning rate 0.1)

- SVM: RBF kernel (gamma auto)

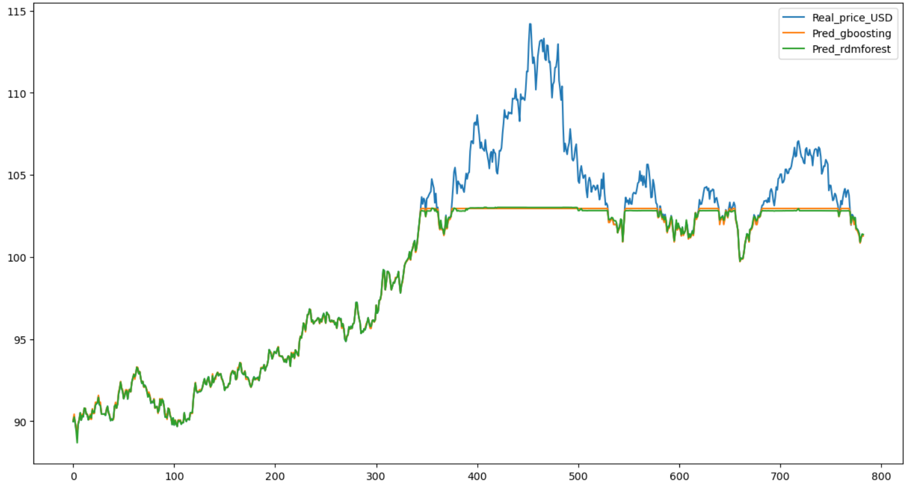
Key question

Which model survives the Fed policy shock?

Graph of Gradient Boost and Random Forest on 3M interest rate (USD3MD=)



Graph of Gradient Boost and Random Forest on USD index (=USD)



The regime shift

Federal Reserve tightening (March 2022)

Event: 0% to 4.25% (50bp hikes $\times$ 9)

All ML models failed

From Dec 2021 $\rightarrow$ observation 200 onward:

- Actual rates: increase 4% (sharp tightening)
- No model captured this sharp increase accurately

Why models failed

Root Cause 1: Non-stationarity

Training data (2014-2020): Rates $= 0.25\% +/- 3\%$ (trending, range)
Test data (2020-2024): Rates $= 0\% \rightarrow 5.5\%$ (trending, non-stationary)
Problem: Models learned low-volatility regime, couldn't generalize to rate shock

Root cause 2: Exogenous policy shift

Fed policy not in feature set (no macro indicators)
Problem: Model trained on historical correlations; Fed tightening is exogenous $=$ breaks all models

Root cause 3: Insufficient history

No 2008 crisis, no 1990s tightening cycles in training
Problem: Models have never seen rate shock of this magnitude

USD 3M rate forecasting: Accuracy comparison

Model	MAE	RMSE	Rank
Random Forest	**1.215**	**2.194**	#1
GBoost	1.517	2.564	#2
SVM	3.193	3.861	#3
MLP	3.443	3.865	#4
LSTM	7.027	8.634	#5

LSTM shock: 7.027 MAE = predicting -700bp when actual is +400bp on 10bp move scale

Implication

LSTM (most complex) is **5.8 worse** than Random Forest (simpler)
This is the opposite of what deep learning advocates promised

USD FX index forecasting: Similar pattern

Model	MAE	RMSE	Rank
Random Forest	**1.354**	**2.194**	#1
GBoost	1.609	2.348	#2
LSTM	2.507	3.049	#3
MLP	2.505	3.049	#4
SVM	19.838	19.838	#5

Same pattern: Tree-based methods (RF, GBoost) dominate neural networks

But: SVM completely fails on FX

Key insight

Random Forest performs best on **BOTH** targets = robust to regime shifts better than neural networks

Random Forest advantage #1: Non-Parametric

Neural Networks

- Assume data follows learned distribution
- Parameters capture training dynamics
- Regime shift = parameters become invalid
- Result: extrapolate confidently into wrong regime

Random Forest

- No distributional assumption
- Simply averages similar historical trees
- New regime = use closest trees
- Result: conservative, don't extrapolate

Consequence

When Fed tightens (unprecedented in training set), RF believes "no opinion" (stays flat).
LSTM believes "I learned the pattern" (confidently wrong).

Random Forest advantage #2: Local learning

Random Forest uses local neighborhoods

When	RF finds	LSTM
Rates 0-1%	Similar historical trees	learned pattern
Rates 1-2%	Different trees activate	same learned weights
Rates 3-5%	Extrapolation: sparse trees	confidently wrong

Example: Rate jumps from 0.5% $\rightarrow$ 4.5%

RF: No trees trained at 4.5%, limits prediction

LSTM: Learned 0-0.5% pattern, extrapolates to 4.5% = **massive error**

Takeaway

"Curse of extrapolation": Neural nets extrapolate confidently; trees conservatively

Random Forest advantage #3: Automatic feature selection

Neural Networks

- All 20 features = same weight
- Learned weights same across all data
- Cannot adapt to regime
- Fed policy absent from features = disaster

Random Forest

- Each tree uses subset of features
- Different trees prioritise different features
- High-vol regime trees use different features
- Result: adaptive to regime

Implication

To fix LSTM: Add Fed Funds Rate + CRB Commodity Index + Volatility Index (VIX)
To use RF well: Nothing needed, already adapts

Three insights for PMs

Use simple models for regime-switching markets

Financial time series have exogenous shocks (Fed policy, geopolitics, earnings surprises)

Neural networks assume stationary patterns

Solution: Random Forest + switching rules (detect VIX $> 30 \rightarrow$ reduce weight on ML)

Add macro indicators or accept conservatism in forecasting

LSTM failed because it had NO Fed-related features

Combination of different model types

Combine: **Random Forest + GBoost** with calibrated weighting = best of both $\rightarrow$ reduces single-model catastrophe risk

For traders

If using LSTM/MLP for rates:

1. Add Fed policy detector (alert when CRB $>$ 90th %ile)
2. Override predictions during crises
3. Reduce position sizes when model confidence high but market unusual

If using Random Forest:

1. Trust predictions in quiet periods (rate range-bound)
2. Remember RF won't capture Fed shocks $=$ accept 50-100bp forecast error
3. Combine with fundamental view (Fed-watching)

For USD FX specifically:

RF performance best, but SVM collapsed
Lesson: Not all ML models generalizable across assets
Test each model on SPECIFIC asset before deploying

Known limitations

Limitation 1: Missing features

No macro indicators (Fed Funds, VIX, CRB, credit spreads)
Adding these would improve LSTM significantly

Limitation 2: Limited training on multiple crisis

Training data (2018-2022): Only taper tantrum + pandemic
No 2008 financial crisis, no 1990s tightening
Would 2008 training data help? Unknown.

Limitation 3: Single regime shift

Fed tightening 2022-2024 = 1 crisis sample
Does RF work better for other shocks? Need more data.

Limitation 4: Hyperparameter tuning

RF used defaults (50 trees)
Intensive tuning might improve all models → overfitting bias?

Future research

Research direction 1: Feature engineering

Add Fed Funds Rate + VIX + Credit Spreads + Unemployment
Retrain LSTM with macroeconomic features
Expected: LSTM improves to RF-level performance

Research direction 2: Adaptive models

Build model that detects regime shifts
Switch LSTM $\leftrightarrow$ RF based on VIX threshold
Test on 2008, 2011, 2020 crises

Research direction 3: Ensemble learning

Combine LSTM (captures trends) + RF (ignores noise)
Use volatility regime to weight models dynamically
May beat either model alone

Bottom line

Four takeaways

1. Random Forest outperforms LSTM by 66-83% on financial time series with regime shifts
2. Neural networks fail catastrophically during exogenous policy changes (Fed tightening)
3. Simple non-parametric methods (RF, GBoost) more robust than deep learning to market shocks
4. For rates/FX forecasting: Use RF unless you have macro indicators to feed LSTM

For the industry

The ML hype cycle has oversold neural networks for finance
Simpler, less sexy methods often work better
Add macro features or accept conservative predictions

Questions?

Youssef Louraoui
ESSEC Business School
youssef.louraoui@essec.edu

Backup: Model architecture details

Model	Architecture	Key Parameters
MLP	3-layer perceptron	Dropout 25%, epoch 10, batch 16
LSTM	Sequence-to-sequence	Lookback 30, dropout 25%, epoch 10
Random Forest	Ensemble of 100 trees	Max depth 20, min samples 5
GBoost	Sequential boosting	Learning rate 0.1, depth 5
SVM	RBF kernel	Gamma auto, C 1.0

Key finding

Even with regularisation (dropout 25%), LSTM couldn't handle Fed shock
Problem wasn't overfitting but model type mismatch for regime-switching data

Backup: Feature engineering ideas

To improve LSTM performance, add:

Feature	Rationale
Federal Funds rate	Direct policy indicator
2-Year Yield	Market expectations of path
VIX (Volatility Index)	Risk-off triggers regime shifts
Credit spreads (HY OAS)	Financial conditions indicator
Unemployment Rate	Dual mandate for Fed
CRB Commodity Index	Inflation proxy
Trade-Weighted USD	External factor for FX
PCE Inflation	Fed's target metric

Hypothesis: With these features, LSTM MAE might drop from 7.0 to 2.5-3.0 (closer to RF)

Backup: Hyperparameter tuning impact

Potential improvements through tuning:

- **LSTM:** Add attention mechanism (captures long-range dependencies)
- **MLP:** Increase layers (4-5 layers) + learning rate schedule
- **RF:** Increase trees (500-1000) + test depth 10-30
- **GBoost:** Lower learning rate (0.01-0.05)
- **SVM:** Kernel selection (polynomial vs RBF)

Key insight: Even with aggressive tuning, RF likely remains best performer due to non-parametric nature