



## MACRO STRATEGY

# Applying an Embedded Ridge Model to Forecast the Implied Volatility Surface for Foreign Exchange

May 4, 2020

### Summary

- Forecasting implied FX volatility surface has important applications in investment and trading, but it is a challenging task because the FX vol surface is composed of correlated high-dimensional time series and the shape of the surface can be significantly impacted by extreme market shocks that are almost impossible to forecast.
- Vector Autoregression (VAR) has been utilized often in forecasting multi-variate time series, but has limitations due to high dimensional highly correlated data, which exists for this use case.
- We designed a new way to regularize the VAR model to improve its accuracy by training a regularized linear regression model (i.e. Ridge Regression) on the Embedded dataset, which we called Embedded Ridge (ER) model.
- The Embedded Ridge model was applied to forecast 8 FX implied volatility surfaces for 1 day, 1 week and 1-month forecast horizons.
- Our results showed that the Embedded Ridge model outperformed the VAR model in the measure of accuracy, computations time and stability.

## 1. INTRODUCTION

Forecasting of FX implied volatility is important for portfolio management including FX hedging, trading and developing an investment strategy. A reasonable forecast of the FX implied volatility surface will help traders choose which term to use and when to hedge. In this research series, we conducted our research with the following three questions in mind:

- What are the economic and financial factors that impact the most on FX implied volatilities?
- Can we apply machine learning algorithms to forecast FX implied volatility surface, in order to outperform the traditional econometric models?
- Are we able to increase forecast confidence by doing a triangle analysis, i.e. levels, direction and rank forecasts? On this question, the idea is that if level forecast, direction forecast and rank forecast all agree with each other, our confidence in our forecast should be relatively high.

In the first paper of this series, we applied an Embedded Ridge algorithm to forecast implied volatility surfaces of eight FX pairs. In the following papers, we will apply other machine learning algorithms to forecast level, direction and rank of the implied volatility surfaces of these FX pairs.

Forecasting high-dimensional data is challenging because of the many possible attribute combinations that need to be forecasted. To address this issue, some researchers design new models that select subsets of lags and dimensions by cross validation, variable selection algorithms or by regularization. Regularizing time series is well studied, for example [1], [2], [3] and more. We present here a new way to improve the accuracy in forecasting time series, via a reduction of the output of the autoregressive process to the embedded dataset and regularizing it. Regularization is particularly useful to mitigate the problem of multicollinearity and ill-conditioning (ill-posed). It is well known that instability of solutions to small changes in inputs causes many problems in numerical computations. Existence, uniqueness and stability of solutions are important features of mathematical problems. Problems that fail to satisfy these conditions are called ill-posed [4]. In general, Regularization provides improved efficiency in parameter estimation problems in exchange for a tolerable amount of bias [5]. Embedding is a horizontal concatenation of the lags of time series. This way, any

model, such as linear regression, can forecast the time series from the embedded dataset. In this paper, we used embedded linear regression with ridge regularization to forecast the eight FX implied volatility surfaces.

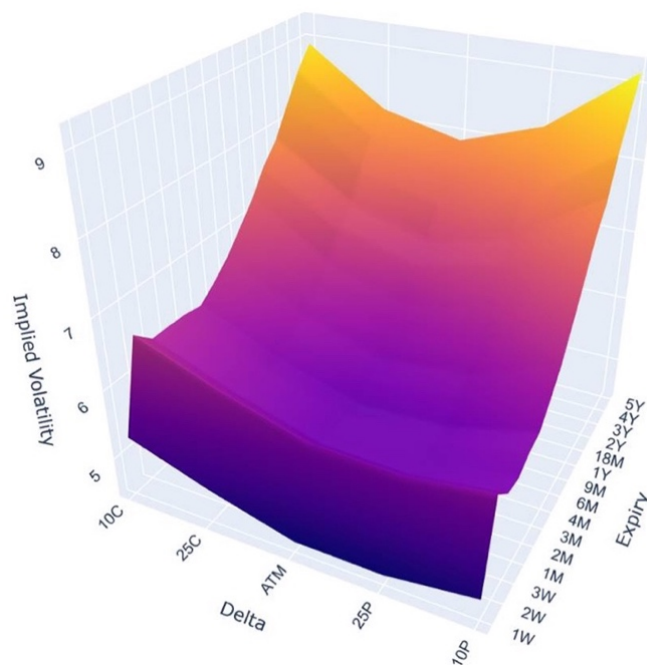
## 2. DATA

The data used in this study consisted of multiple time series taken from eight volatility surfaces of currencies traded on FOREX: AUDUSD, USDMXN, GBPUSD, USDJPY, USDBRL, USDCAD, EURUSD, USDKRW.

The data was taken from Bloomberg and is from January 1, 2014 through June 11, 2019, a total of 1420 days. Each surface consisted of 75 time-series (later, “points”), except for the USDJPY and USDKRW sets which consisted of 69 and 45 points, respectively. The 75 points corresponded to five values of delta: 10 Call, 25 Call, At the Money, 25 Put and 10 Put, and fifteen values of expiry: 1, 2, and 3 weeks, 1, 2, 3, 4, 6, and 9 months, 1, 1.5, 2, 3, 4 and 5 years out.

An example of constructed surface can be seen in Figure 1.

**Figure 1 | Constructed surface of EURUSD of June 11, 2019**



The missing points in USDJPY and USDKRW were imputed from the nearest points as follow: We first forecast the surface using the given points and later

interpolate and extrapolate the missing points from the forecast. In case of missing points between two points, we used weighted average. So for example, if with delta 10P, the expiries 2M, 3M are missing, we assigned 1/3 of the difference between 1M and 4M to 2M and 2/3 of it to 3M. In case of missing at the edge of the surface, we extrapolated the missing value from the nearest point. So for example, if with delta 10P, the expiries 4Y and 5Y are missing, we calculated the ratio between 4Y and 3Y to be the same as between 3Y and 2Y. Once we calculated 4Y, we calculated the 5Y with the same ratio from 4Y.

### 3. METHODOLOGY

Forecasting future levels of time series is commonly done by Auto Regressive Moving Average (later, **ARMA**) models for univariate time series, and Vector Auto-Regressive (later, **VAR**) models for multivariate interdependent time series.

Since our dataset contains multiple time series which are highly interdependent, we compared our new model to **VAR**. We trained our models on each pair separately.

#### 3.1. Multicollinearity Problem and Regularization

Points on FX volatility surface are highly correlated. So, any linear model that will use these points as predictors to other points will have the multicollinearity phenomenon. Multicollinearity or collinearity is the existence of near-linear relationships among the regressors, predictors, or exogenous variables. Multicollinearity or ill-conditioning can create inaccurate estimates of the regression coefficients, inflate the standard errors of the regression coefficients, deflate the partial t-tests for the regression coefficients, give false and nonsignificant p-values, and degrade the predictability of the model. It also causes changes in the direction of signs of the coefficient estimates [6]. One common way to control the multicollinearity in the model is to use regularization. Regularization technics involve removing or reducing the coefficient of correlated variables. See [7] for examples of multicollinearity and regularization techniques to control it.

In this work we trained two models, VAR and Embedded Ridge (ER). The first was trained without regularization, and the second was trained with regularization. We found, as expected, that regularized model performed better than the unregularized model.

#### 3.2. VAR (Vector Auto-Regression)

**VAR** is a stochastic process model used to capture the linear interdependencies among multiple time series. **VAR** models generalize the univariate autoregressive model (AR model) by allowing for more than one evolving variable. All variables in a **VAR** enter the model in the same way: each variable has an equation explaining its evolution based on its own lagged values, the lagged values of the other model variables, and an error term.

A **VAR** model describes the evolution of a set of  $k$  variables over the same sample period ( $t = 1, \dots, T$ ) as a linear function of only their past values.

A  $p^{th}$  order **VAR**, denoted **VAR**( $p$ ), is

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1 \mathbf{y}_{t-1} + \mathbf{A}_2 \mathbf{y}_{t-2} + \dots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{e}_t$$

where the observation  $\mathbf{y}_{t-i}$  is called the  $i^{th}$  lag of  $\mathbf{y}$ ,  $\mathbf{c}$  is a  $k$ -vector of constants (intercepts),  $\mathbf{A}_i$  is a time-invariant ( $k \times k$ )-matrix and  $\mathbf{e}_t$  is a  $k$ -vector of error terms.

We trained our **VAR** models on multiple hyperparameters using cross validation. The hyperparameters that achieved best performance was: **VAR** with 5 lags, i.e. **VAR**(5), and without constants.

For clarification, in our work,  $\mathbf{y}$  represents a vector that contains 75 time series, and  $\mathbf{y}_t$  represents, its temporal representation to date  $t$ .

The residuals  $\mathbf{e}_t$  are:

$$\mathbf{y}_t - \mathbf{A}_1 \mathbf{y}_{t-1} - \mathbf{A}_2 \mathbf{y}_{t-2} - \dots - \mathbf{A}_5 \mathbf{y}_{t-5}$$

And they satisfy:

$$E(\mathbf{e}_t) = \mathbf{0}$$

$$E(\mathbf{e}_t \mathbf{e}'_t) = \text{Constant}$$

$$E(\mathbf{e}_t \mathbf{e}'_{t-k}) = \mathbf{0}$$

From the above discussion, **VAR**( $p$ ), models dataset with dimension of  $k \times p \times t$ . In our dataset  $75 \times 5 \times t = 375 \times t$ .

Since the order of integration of the data is 1, we differenced the time series by:

$$\mathbf{y}_t - \mathbf{y}_{t-1}$$

This transformation only applied to **VAR** model.



### 3.3. Reduction

In computability theory and computational complexity theory, a reduction is an algorithm for transforming one problem into another problem. We will use the following embedding algorithm to *reduce* the *VAR* problem into Linear Regression problems.

### 3.4. Embedding

Embedded dataset is a horizontal concatenation of previous lags.

More formally, given dataset  $Y$  of size  $(N \times M)$ , i.e.  $N$  samples of  $M$  time series, we want to create a new dataset  $Y_K^M$  of size  $(N \times KM)$ , i.e.  $N$  samples of  $KM$  time series. We can do so by creating new  $K$  datasets from  $Y$ . Each dataset is one of  $K$  lags of  $Y$ . Then by concatenating horizontally the  $K$  datasets we will get our predictors' dataset. The original  $Y$  will be our target dataset.  $Y_K^M$  is the *predictors* dataset.

The matrix representation of  $Y_K^M$  is:

$$Y_K^M = \begin{bmatrix} y_{t-1}^{m=1} & y_{t-2}^{m=1} & \dots & y_{t-K}^{m=1} & y_{t-1}^{m=2} & \dots & y_{t-K}^{m=M} \\ y_{t-2}^{m=1} & \dots & \dots & y_{t-K-1}^{m=1} & y_{t-2}^{m=2} & \dots & y_{t-K-1}^{m=M} \\ \vdots & \ddots & \dots & \vdots & \vdots & \ddots & \vdots \\ y_{t-n}^{m=1} & \dots & \dots & y_{t-K-n}^{m=1} & y_{t-n}^{m=2} & \dots & y_{t-K-n}^{m=M} \end{bmatrix}$$

To create the *respond* dataset (column) we need to consider the number of steps of the forecast. Let  $S$  be the number of steps,  $Y'$  the respond and  $m$  will be the time series we want to forecast. Then, given some  $S, m$ :

$$Y' = \begin{bmatrix} y_{t-1+S}^m \\ y_{t-2+S}^m \\ \vdots \\ y_{t-n+S}^m \end{bmatrix}$$

### 3.5. Regularization

Either *VAR* vanilla model and linear embedding model creates dataset with high dimensions  $O(mk)$ . More than that, the surface's vectors are highly correlated, either between themselves or autocorrelated between their lags. These properties turn our new embedding problem into an "*ill-posed problem*". A common solution to this is using regularization.

In our study we deployed Ridge regularization which defined as follow:

When learning a linear function  $f$ , characterized by an unknown vector  $\beta$  such that  $f(x) = \beta \cdot x$ , one can add the  $L_2$ -norm of the vector  $w$  to the loss expression in order to prefer solutions with smaller norms. It is expressed as:

$$\min \sum_{i=1}^n L(\beta; y, x) + \lambda R(\beta)$$

### 3.6. Embedded Ridge (ER)

We trained Linear Regression with Ridge Regression over the embedded dataset, defined as:

$$\min \sum_{i=1}^n L(\hat{x}_i \cdot \beta, \hat{y}_i) + \lambda \|\beta\|_2^2$$

In each iteration, we picked one column from  $X$  as a target and all columns in  $X_k$  as our predictors.

As required by Ridge, the data was normalized before. No other data preparation was needed.

### 3.7. Cross-Validation and testing

For training, cross-validation and hyperparameters selection we used the dataset from January 1, 2014, through January 30, 2018, a total of 1065 days. For testing, we used a dataset from January 31, 2018 to June 11, 2019 total 355 days. So, 75% went for training and validation and 25% for testing.

Testing time series models is different from other models and required different procedure. Obviously, training can only be done with *past* dataset and testing only on *future* dataset. That requires to test on series of test sets, each consisting of a *single* observation. For each test, the corresponding training set consists only of observations that occurred prior to the observation that forms the test set.

So, first train dataset contained 1065 consecutive observations from January 1, 2014 to January 30, 2018 and one observation January 1, 2014 for testing. Second train dataset contained 1066 observation (January 1, 2014 to January 1, 2014) one observation (February 1, 2018) for testing. Last day of testing was June 11, 2019.

This way we forecasted 355 values. Later we summarized these 355 forecasts.

To forecast 5 and 21 steps, we followed the same procedure, except for the testing sample which was 5 or 21 days ahead.

## 4. RESULTS

### 4.1. Models Hyperparameters

As we have said above in 3.6, we chose the **VAR** and **ER** models' hyperparameters at the training and cross-validation part. Best parameters for VAR on the training dataset was an order of 5, i.e. 5 lags, and no trend. Accordingly, the embedding's parameters were chosen with 5 lags and no constant (no intercept). For the Ridge part, we chose the  $\lambda$  dynamically, i.e. at each iteration, best  $\lambda$  was chosen through cross-validation, and that best  $\lambda$  was used to forecast future values.

### 4.2. Metrics and Forecast Steps

We forecasted for 1, 5 and 21 days ahead. We measured our performance with Root Mean Square Error (RMSE), as it closely represents the standard deviation of the errors.

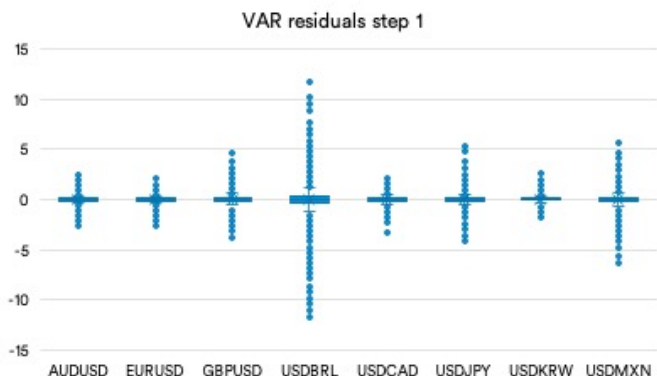
$$RMSE = \sqrt{\frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{T}}$$

First, we will evaluate the quality of residuals. We would like to check whether they follow a *white noise process*. Later, we will evaluate the forecasting performance.

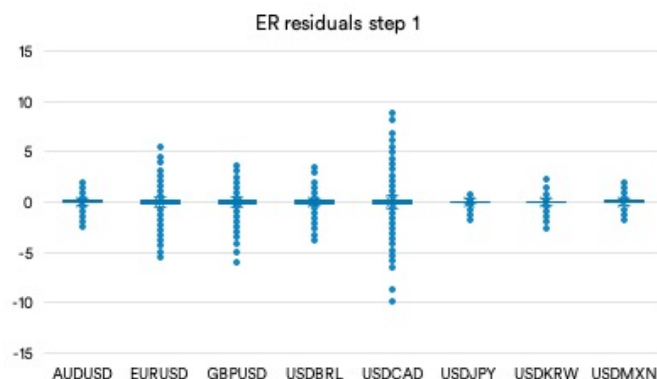
### 4.3. Residuals basic statistics

We present 4 box plots. Two in higher resolution range (-15,15) and two with lower resolution (-1,1).

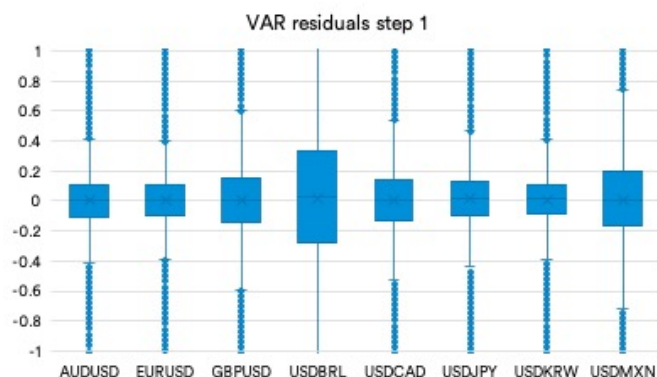
**Figure 2 | VAR residuals are around 0 but hold high range.**



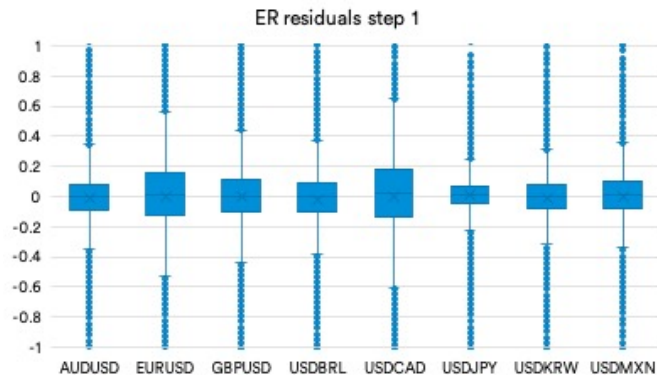
**Figure 3 | ER residuals also around 0 and has less range than VAR has.**



**Figure 4 | Closer look on VAR residuals we can see the higher range**



**Figure 5 | ER has shorter range**



From the box plots of the residuals we can see that both models centred around 0, while ER has lower ranges across all pairs. More precisely the next table show that ER is closer to 0 with less variance around the mean.

## Residuals statistics 1 step all pairs

	ER	VAR
Mean	-0.0038	0.0066
Standard Error	0.0008	0.0010
Median	0.0060	0.0087
Standard Deviation	0.3553	0.4689
Sample Variance	0.1262	0.2198
Kurtosis	68.6	65.4
Skewness	-1.7	-1.3
Range	19.2	23.5
Minimum	-9.9	-11.7
Maximum	9.3	11.8
Sum	-761	1334
Count	202269	202269

Interesting property that we can learn from these statistics is that ER and VAR model are close.

#### 4.4. Durbin Watson test

In order to check for auto correlation between the residuals we run the Durbin Watson test.

The Durbin–Watson (DW) statistic  $d$  is defined as a test statistic used to detect the presence of autocorrelation at lag 1 in the residuals (prediction errors) from a regression analysis. The value of  $d$  always lies between 0 and 4.

$$d = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2}$$

$d = 2$  indicates no autocorrelation. If  $d > 2$ , successive error terms are negatively correlated. If  $d < 2$ , successive error terms are positively correlated.

For each model (VAR and ER), we run the DW test over all pairs and all points. For each point (75 points), we calculated the average of DW over all pairs. We created histograms that describe the distribution of the DW tests results of the 75 points.

Figure 6 | DW for ER 1 step. Close to 2.0 from below. Small positive autocorrelation between residuals.

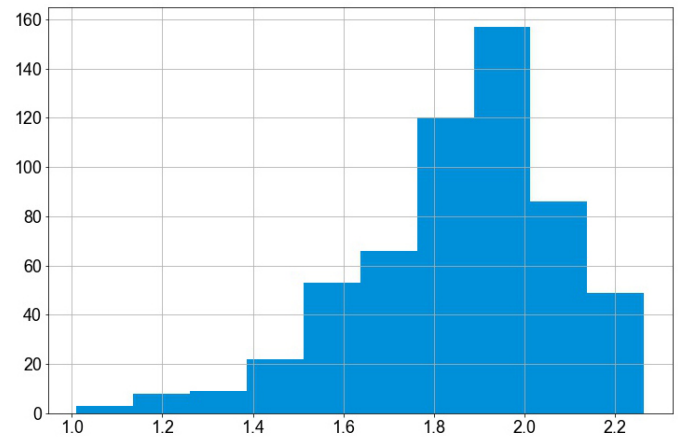


Figure 7 | DW for VAR 1 step. Close to 2.0 from up. Small negative autocorrelation between residuals.

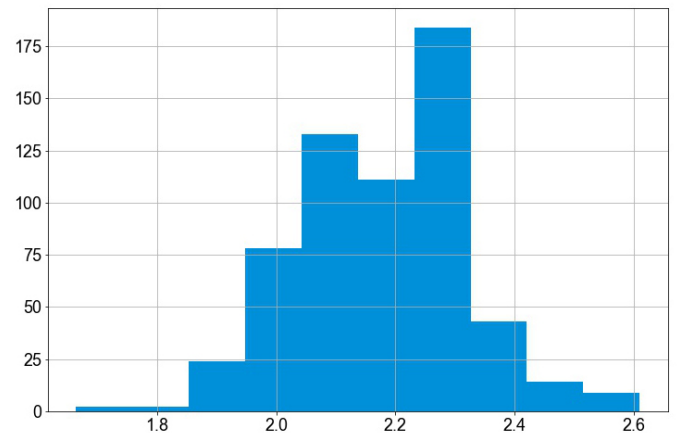


Figure 8 | DW for ER 5 step

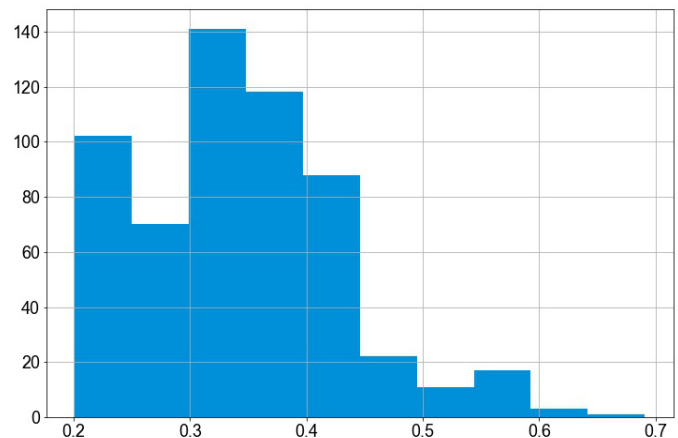


Figure 9 | DW for VAR 5 step

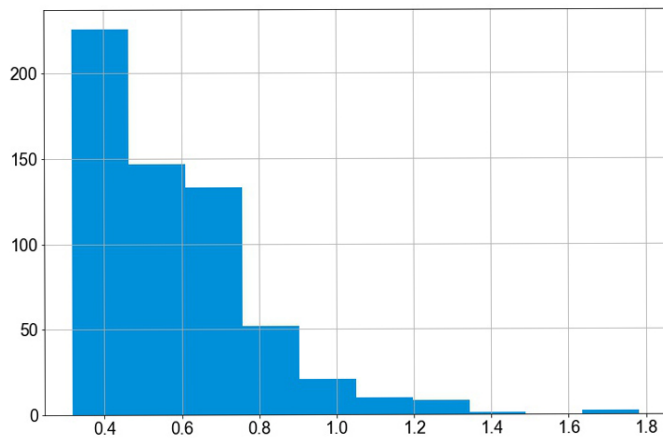


Figure 10 | DW for ER 21 step

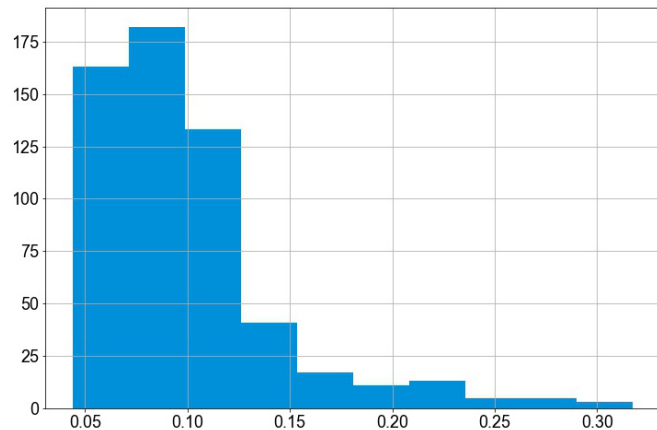
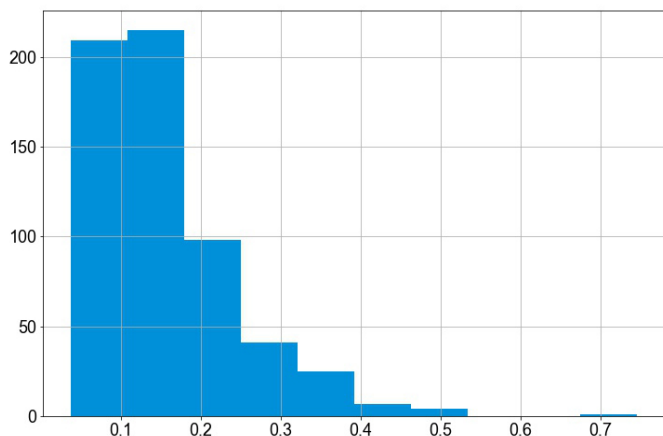


Figure 11 | DW for VAR 21 step



Durbin Watson tests show for step 1 minor auto correlation, while strong auto correlation in 5 and 21 steps.

Durbin Watson autocorrelation test

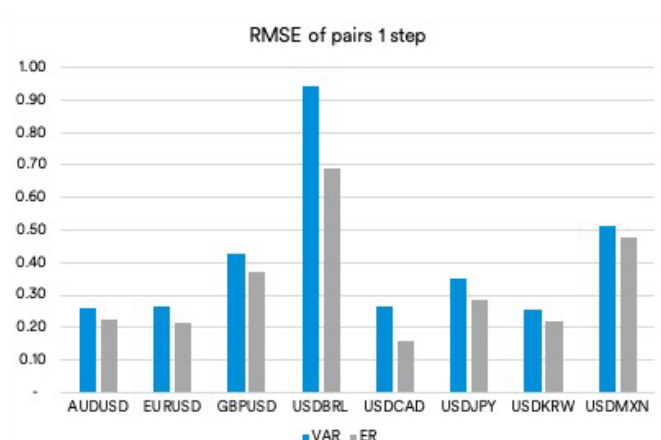
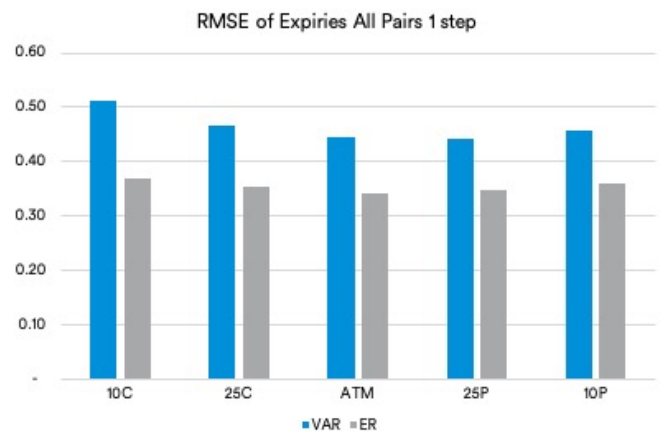
step	model	DW
1	ER	1.848302
	VAR	2.178545
5	ER	0.34423
	VAR	0.578055
21	ER	0.098533
	VAR	0.153603

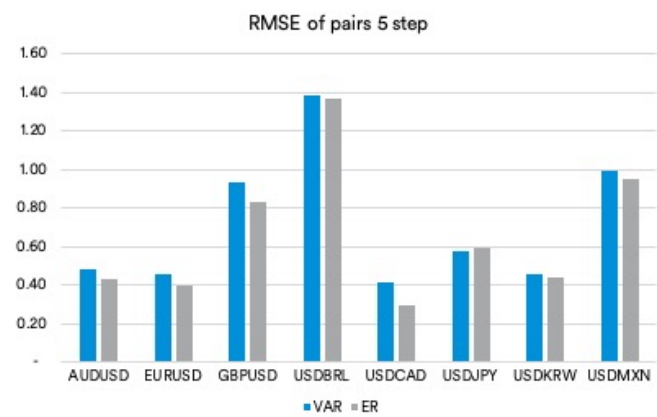
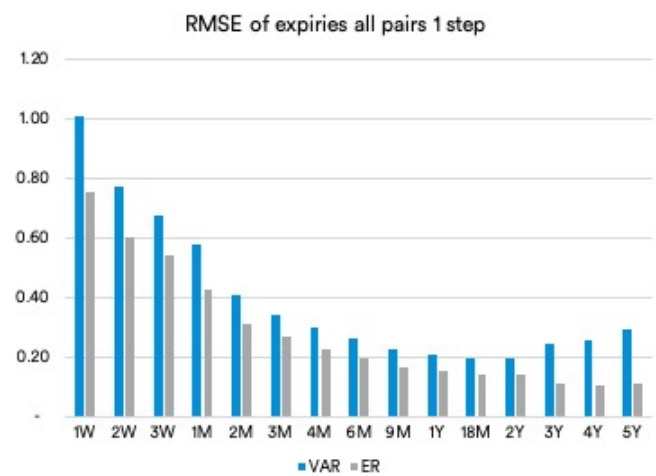
ER presents slightly better results in DW test for 1 step.

#### 4.5. Out-of-Sample Performance

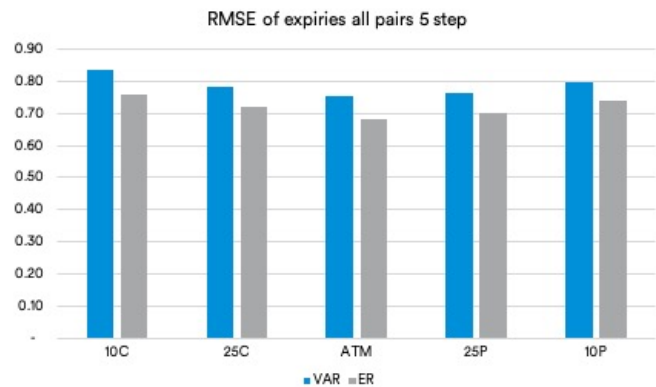
We measured the performance of the models with RMSE.

For 1 step:

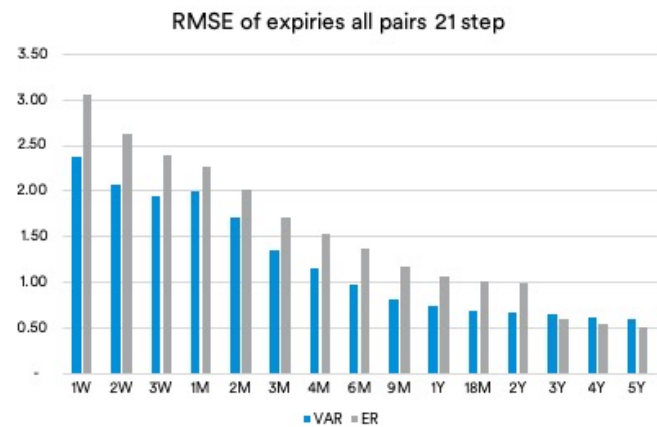
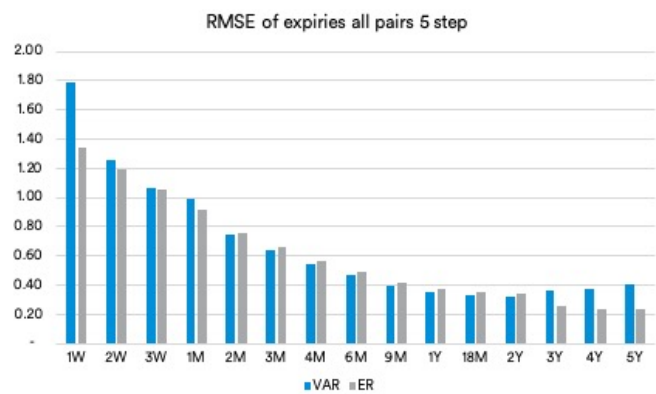
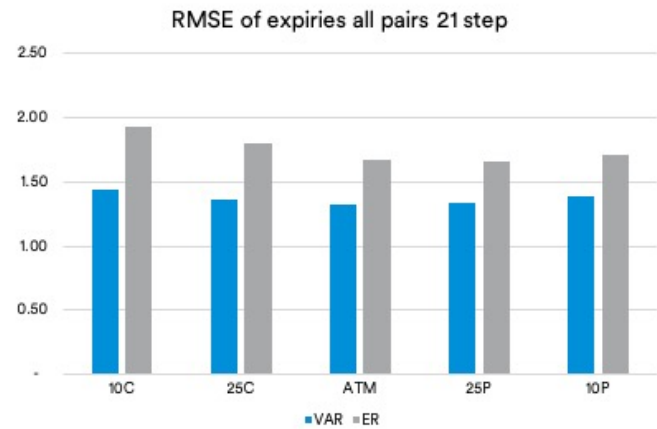




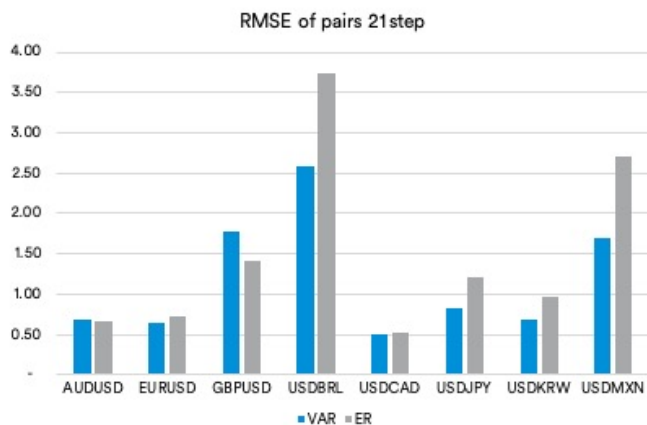
For 5 steps:



For 21 steps:







ER performance is better in forecasting 1 and 5 days ahead, but less accurate in forecasting 21 days ahead.

#### 4.6. Run Time

Embedded Ridge evaluate 355 evaluations in less than 9 seconds, while VAR needs more than 30 seconds to do the same evaluation task. That makes the ER model 3.5 times faster while working on the same dataset (values and size).

## 5. CONCLUSION

We presented a new way for a regularized linear model to learn from time series. That been done through reduction of the AR process with embedding.

We tested the embedding on FOREX volatility surface and found that the reduced problem, was performing better for short horizons and in less time than VAR.

## 6. FURTHER WORK

Time series contains two processes, autoregressive (AR) and moving average (MA). In our work, we used embedding to reduce the autoregressive (AR) part of time-series. Later work might use embedding to reduce the moving average (MA).

Later work might also forecast with other models: (other than Ridge Regression) linear and nonlinear, parametric and non-parametric.

The DW results (subsection 4.4), show that linear models (VAR and ER) that we used in this work have difficult to forecast 5 and 21 steps ahead. We think that Temporal Convolution Neural Network, or Error Correction Model

might be a good alternative candidate for forecasting many steps in the future, see [8] and [9].

We also used only consecutive lags (from 1 to 5) to forecast future values. Further work can leverage the embedding by using non-consecutive lags such as, 1st lag, 5th lag and 6th lag.

## 7. REFERENCE

- [1] Nicholson, W., Matteson, D., & Bien, J. (2017). Bigvar: Tools for modelling sparse high-dimensional multivariate time series. arXiv preprint arXiv:1702.07094.
- [2] Davis, R. A., Zang, P., & Zheng, T. (2016). Sparse vector autoregressive modelling. *Journal of Computational and Graphical Statistics*, 25(4), 1077-1096.
- [3] Wilms, I., Basu, S., Bien, J., & Matteson, D. S. (2017). Sparse Identification and Estimation of Large-Scale Vector AutoRegressive Moving Averages. arXiv preprint arXiv:1707.09208
- [4] Öztürk, Fikri, and Fikri Akdeniz. "Ill-conditioning and multicollinearity." *Linear Algebra and Its Applications* 321.1-3 (2000): 295-305.
- [5] Gruber, Marvin (1998). *Improving Efficiency by Shrinkage: The James–Stein and Ridge Regression Estimators*. Boca Raton: CRC Press. pp. 7–15. ISBN 0-8247-0156-9.
- [6] Saleh, A. M. E., Arashi, M., & Kibria, B. G. (2019). *Theory of Ridge Regression Estimation with Applications* (Vol. 285). John Wiley & Sons.
- [7] Deanna S. G., Henry M. J. F. (2018) *Regulation Techniques for Multicollinearity: Lasso, Ridge, and Elastic Nets*
- [8] Tsantekidis, A., Passalis, N., Tefas, A., Kannianen, J., Gabbouj, M., & Iosifidis, A. (2017, July). Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics (CBI)* (Vol. 1, pp. 7-12). IEEE.
- [9] Engle, R. F., & Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, 251-276.

## Authors

Erez Meoded, Ben Perryman, Jun Jiang, Alex Chien, Caleb Weaver, Al Kirk

# About MetLife Investment Management

MetLife Investment Management (MIM),<sup>1</sup> MetLife, Inc.'s (MetLife's) institutional investment management business, serves institutional investors by combining a client-centric approach with deep and long-established asset class expertise. Focused on managing Public Fixed Income, Private Capital and Real Estate assets, we aim to deliver strong, risk-adjusted returns by building tailored portfolio solutions. We listen first, strategize second, and collaborate constantly as we strive to meet clients' long-term investment objectives. Leveraging the broader resources and 150-year history of the MetLife enterprise helps provide us with deep expertise in navigating ever changing markets. We are institutional, but far from typical.

For more information, visit: [metlife.com/investmentmanagement](https://metlife.com/investmentmanagement)

<sup>1</sup> MetLife Investment Management ("MIM") is MetLife, Inc.'s institutional management business and the marketing name for the following affiliates that provide investment management services to MetLife's general account, separate accounts and/or unaffiliated/third party investors: Metropolitan Life Insurance Company, MetLife Investment Management, LLC, MetLife Investment Management Limited, MetLife Investments Limited, MetLife Investments Asia Limited, MetLife Latin America Asesorias e Inversiones Limitada, MetLife Asset Management Corp. (Japan), and MIM I LLC.

---

## Disclosure

This material is intended for institutional investor and professional investor use only. Not for use with retail public.

This document has been prepared by MetLife Investment Management ("MIM") solely for informational purposes and does not constitute a recommendation regarding any investments or the provision of any investment advice, or constitute or form part of any advertisement of, offer for sale or subscription of, solicitation or invitation of any offer or recommendation to purchase or subscribe for any securities or investment advisory services. The views expressed herein are solely those of MIM and do not necessarily reflect, nor are they necessarily consistent with, the views held by, or the forecasts utilized by, the entities within the MetLife enterprise that provide insurance products, annuities and employee benefit programs. The information and opinions presented or contained in this document are provided as the date it was written. It should be understood that subsequent developments may materially affect the information contained in this document, which none of MIM, its affiliates, advisors or representatives are under an obligation to update, revise or affirm. It is not MIM's intention to provide, and you may not rely on this document as providing, a recommendation with respect to any particular investment strategy or investment. The information provided herein is neither tax nor legal advice. Investors should speak to their tax professional for specific information regarding their tax situation. Investment involves risk including possible loss of principal. Affiliates of MIM may perform services for, solicit business from, hold long or short positions in, or otherwise be interested in the investments (including derivatives) of any company mentioned herein. This document may contain forward-looking statements, as well as predictions, projections and forecasts of the economy or economic trends of the markets, which are not necessarily indicative of the future. Any or all forward-looking statements, as well as those included in any other material discussed at the presentation, may turn out to be wrong.

L0520003595[exp0522][All States] L0520003591[exp0522][All States] L0520003540[exp0422][All States] L0520003602[exp0422][All States]

