Hybrid Autoregressive-Recurrent Neural Network Architecture for Algorithmic Trading of Cryptocurrencies

Persson, Samuel `joelpe@stanford.edu`
Slottje, Andrew `slottje@stanford.edu`
Shaw, Ian `ieshaw@stanford.edu`

March 22, 2018

Abstract

We investigate deep learning algorithms for cryptocurrency pricing. Previous work in deep learning for time-series data has focused on recurrent neural networks. We consider a recently developed model, the "R2N2," which includes residuals from vector autoregression in the RNN feature set. We improve the computational efficiency of this algorithm and obtain good accuracy for predicting returns on a basket of cryptocurrencies, indicating arbitrage opportunities in the market.

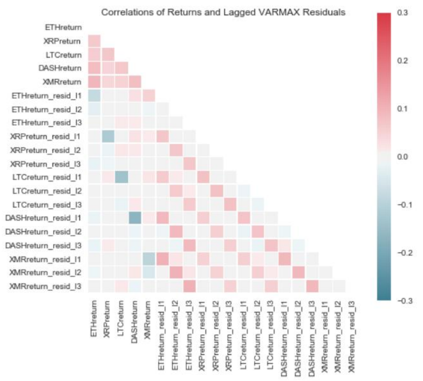
1 Introduction

Time-series models provide the primary set of statistical forecasting methods to address time dependence in feature data. Yet these models often perform poorly - so much so that obtaining accurate time-series forecasts is ranked one of the most important problems in modern data mining [1]. Compounding this problem, market prices update to reflect all available information, so financial-asset time series are particularly challenging to predict: by the "no arbitrage principle," which composes one of the fundamental laws of asset pricing, time dependency must be priced out of the market [2]. This barrier to predictability may explain why the application of neural networks to time series data, which dates from the early 1990s [3], has long been limited to fields like agricultural science and climatology, whose data are not circumscribed in their predictability by arbitrageurs.

Längkvist et al. [4] and Heaton et al. [5] have recently reviewed the literature on neural network prediction of time series and cite the success of long short-term memory networks in this space [6, 7]. A recently developed algorithm, called "R2N2" for "Residual Recurrent Neural Network," first models the time series using vector autoregression (VAR), then employs the resulting residuals as features in a recurrent neural network [8]. The idea is that the VAR extracts linearity in the autocorrelation so that the neural network optimization does not need to extract it and may focus on learning volatility structure. Hybrid models in this style date at least to 2003; however, the vast majority of these applications have again been in non-financial fields [9]. We adapt the R2N2 algorithm to a financial setting using VARMAX modeling, which allows dependence on a higher-dimensional feature set by providing dimensionality reduction in the optimization objective. We perform an architecture and hyperparameter search to build a return-predictive LSTM network that includes residuals from this regression in its feature set. We assess recall, accuracy, and distributional convergence for this model. Our results demonstrate effectiveness of R2N2 models for cryptocurrency pricing and evince presence of excess returns in the cryptocurrency market.

2 Data

We use data uploaded to the internet by Rami Kawach, the founder of the cryptocurrency exchange Bittrex [10]. The data comprise price series in multiple cryptocurrencies as well as trading features such as volume and spread. Our feature set includes hourly return, volume, base volume, and spread in Ethereum (ETH), Litecoin (LTC), Ripple (XRP), Dash (DASH), and Monero (XMR). We consider Bitcoin-denominated prices in order to control for exuberance effects in the cryptocurrency market. These coins



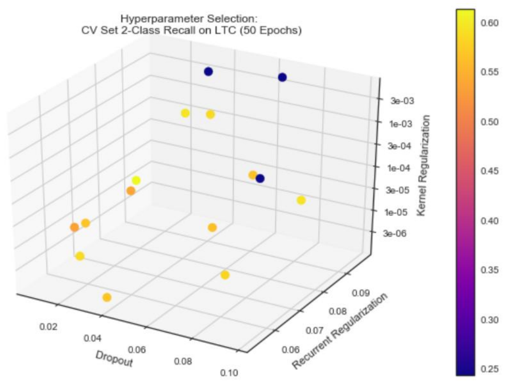


Figure 1: At left, illustration of strong correlation effects in returns and lagged VARMAX residuals. At right, hyperparameter search visualization.

were selected as they were the most highly capitalized five currencies at the beginning of our study period. The duration of our dataset spans September 13, 2015, to June 25, 2017.

To avoid bias with respect to the market regime, we split the data into a training set comprising September 2015 through December 2016 (70%), a cross-validation set comprising December 2016 through March 2017 (15%), and a test set comprising the remainder through June 2017 (15%). Because time-dependent predictions should not incorporate future information, we do not partition the data randomly, as is typical in some other deep-learning applications.

3 Methods

3.1 Vector Autoregression (VAR) and Vector Autoregressive Moving Average Model with Exogenous Variables (VARMAX)

The original R2N2 algorithm makes use of vector autoregression (VAR), a method for modeling the evolution of a multidimensional stochastic process ω_t in discrete time. The VAR estimates a linear relationship between response variables and their lagged values as

$$\omega_t = \sum_{i=1}^{p} A_i \omega_{t-i} + c + \varepsilon_t$$

where A_i define a set of correlation matrices and ε_t gives an error term. We perform AIC and BIC selection to find optimal lag length in the VAR and find optimal lags of 3 and 10 hours, respectively. Cross-validation selects the AIC lag length and gives good results in addition for a one-hour lag, consistent with observed mean reversion in the covariates.

We expand the scope of the R2N2 implementation to incorporate VARMAX modeling. VARMAX specifies an extension of the VAR model by adding moving average (MA) and exogenous (X) regressors: [11]

$$\omega_t = \sum_{i=1}^{p} A_i \omega_{t-i} + Bx_{t-1} + c + \varepsilon_t$$

The specification of exogenous regressors means the model only has to minimize loss in 5 endogenous variables ω_t^i (our selected price returns), and thereby allows a large feature set for the regression at much lower computational cost than in the VAR case. This advantage is important in financial settings, where exogenous variable dependence is often hidden and may span a very high-dimensional space. To our knowledge, this is the first application of VARMAX in this setting. We find statistical non-significance in the moving average terms and choose a lag length of 3 by cross-validation.

3.2 RNN and Residual RNN (R2N2) on VARMAX Residuals

Following the results enumerated above, we opt for LSTM architecture in formulating our RNN implementation. We also calibrate a residual recurrent neural network (R2N2), and we evaluate both VAR and VARMAX-based R2N2 models. The calibration is similar to an ordinary RNN model, except that we include residuals from the vector autoregressive methods in our feature sets. Below, a graph of correlations among lagged VARMAX residuals and observed price returns motivates our use of R2N2.

We create a random architecture search whereby layers of decreasing size (with the order of the nodes, $10^1 \sim 10^2$, based on previous findings in the literature) are randomly added to an LSTM network model with activations randomly chosen as well. We choose the optimal architecture selected by cross-validation, with a moderately deep network providing for the most stably decreasing cost functions on the cross-validation set. Our primary loss metric is mean-squared error, but for purposes of accuracy reporting, we also consider binary cross-entropy loss (we classify the returns in the positive class if they are above .5%). Our results are robust to this change in loss specification. We find optimal RNN architecture of four layers, two of which are LSTM. For R2N2 architecture, we find six LSTM layers to be optimal, a threefold increase. Models are trained with Adam optimization.

Accuracy evaluation of the model selected in this fashion reveals low bias and high variance, so we also calibrate regularization hyperparameters on the cross-validation set by consideration of the cost function shape and recall on both classes. We use a random log-scale search over a window which we refine multiple times. We find optimal dropout parameter ≈ 0.031, optimal recurrent regularization ≈ 0.089, and optimal weight regularization $\approx 7.2 \times 10^{-4}$. A visualization of our hyperparameter search is above; sharp decreases in accuracy for excess weight regularization indicate the importance of log-scale search.

4 Results

We evaluate our models in multiple ways. We report AUC, recall, and accuracy in classification of significantly positive returns and correlation of our signal with observed returns. We tabulate these results by coin for the final model, as well as across a choice of models to evaluate algorithmic efficacy. We also compare distributional convergence of predicted returns and realized returns in a selection of models.

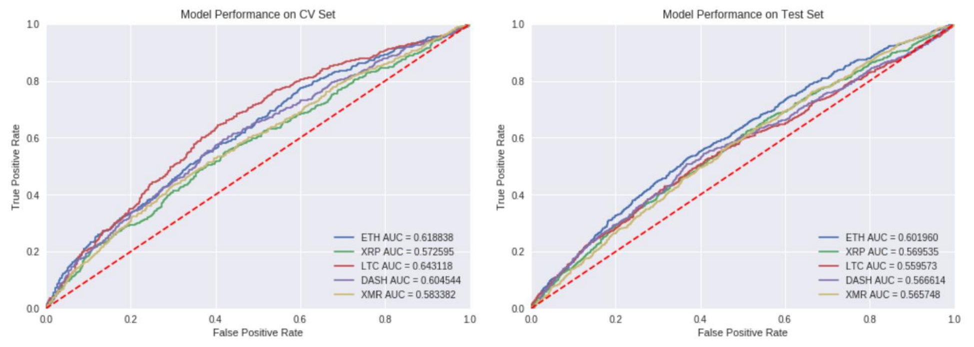


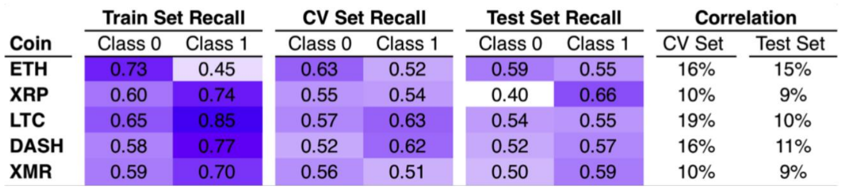
Figure 2: ROC curves on cross-validation and test sets for full currency set under slightly relaxed regularization.

4.1 R2N2 Classification Recall and Returns Correlation

Our R2N2 model achieved high AUC for asset price prediction, with returns classification in all of the coins as high as 60% on the test set. The highest AUC on the test set was Ethereum, which attained 60.2%, while the highest on the cross-validation set was Litecoin, with 64.5%.

To compare performance on the training, cross-validation, and test sets, we considered accuracy within classes (or recall), due to potential for irregularities in the return distribution that can accompany emerging markets. On the test set, our model obtained average recall on both classes ranging from 55% to 57%, with the exception of XRP, where the model obtained good accuracy on the cross-validation set but only obtained 53% average recall on the test set (with especially poor performance on class zero events; this is exactly why we considered recall as our primary metric!). We give a potential explanation for these results below.

Our model was typically able to achieve an improvement of around 10-20% over chance, which is the usual baseline used for financial strategies. Moreover, the typical baseline for correlation in a successful investment signal is 1-2%. We were able to obtain around 10% correlation in all coins, indicating high tradeability of the signal discerned under our model.



	Train Set Recall		CV Set Recall		Test Set Recall		Correlation	
Coin	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	CV Set	Test Set
ETH	0.73	0.45	0.63	0.52	0.59	0.55	16%	15%
XRP	0.60	0.74	0.55	0.54	0.40	0.66	10%	9%
LTC	0.65	0.85	0.57	0.63	0.54	0.55	19%	10%
DASH	0.58	0.77	0.52	0.62	0.52	0.57	16%	11%
XMR	0.59	0.70	0.56	0.51	0.50	0.59	10%	9%

Figure 3: Classification recall for VARMAX-based R2N2 implementation.

4.2 Comparison of R2N2 and Alternative Models

For the purpose of model comparison, we create a baseline strategy, given by even holdings in a basket of the five cryptocurrencies under consideration and prediction of positive returns at every time step. On the set of all coin returns, we find 55% accuracy and 57% AUC for the VARMAX-based R2N2, compared to 50% in both metrics for the baseline. We additionally find the VARMAX-based model shows improvement over the VAR-based model, in line with the enhanced computational efficiency rendered by our improvements, which we expect to improve convergence in the estimate. The basic RNN model does little better than chance, suggesting the residual component of the R2N2 plays a large role in its high accuracy.

	Accuracy	AUC	MSE
R2N2 (VARMAX)	0.550107	0.574483	0.000446
R2N2 (VAR)	0.542687	0.565921	0.000466
RNN	0.513348	0.519892	0.000496
Baseline	0.492196	0.500000	0.999242

Table 1: Comparison of overall accuracy and other selected metrics by model against baseline.

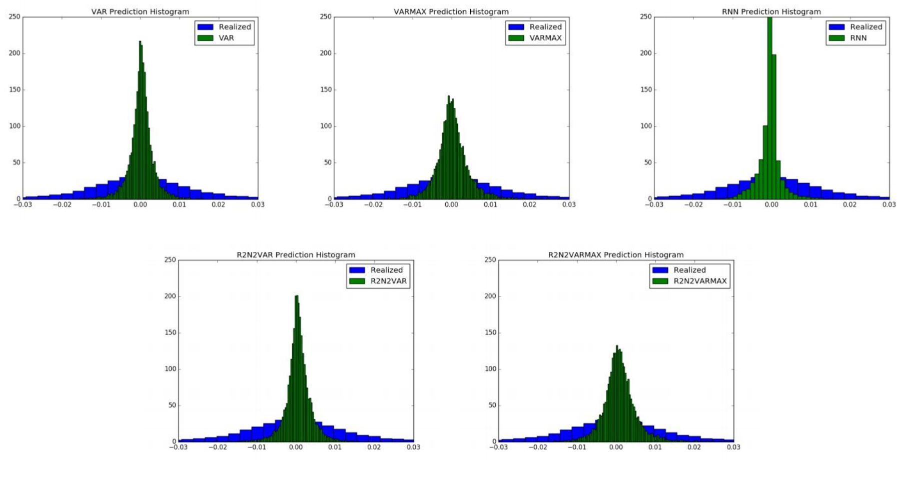


Figure 4: Histograms of predicted returns for each model on the test set superimposed over realized returns demonstrate steady improvement in matching the observed returns distribution.

4.3 Returns Structure in Model Predictions

Finally, we wish to assess whether our model has learned to make reasonable predictions distributionally as well as individually. It is illustrative to compare the returns distribution on both the final model and the models we build on to create the VARMAX-based R2N2, above in 4. We find that the true returns structure is more platykurtic than any of the models, but that in terms of matching the observed returns distribution the VARMAX improves on the VAR and the VAR-based R2N2 improves on the RNN. The closest match is given by the VARMAX-based R2N2. Our algorithmic refinements appear to successively improve the model's ability to learn the problem.

5 Analysis

Because of the no-arbitrage principle, which indicates elimination by arbitrage of anticipated returns, we would expect to find AUC and accuracy of .5 in returns prediction for an efficient market, as well as zero signal correlation, as our baseline obtains. Our single currency 60% test set accuracy and 15% signal correlation strongly indicates evidence of inefficiency in the cryptocurrencies market and presence of arbitrage opportunities.

With respect to the distribution of predicted returns, increased accuracy accompanied by diminution of leptokurticity in our R2N2 predictions as visualized in (4) indicates that the R2N2 model represents an important improvement over an RNN baseline, a proof of concept for future R2N2 applications in the financial setting. Moreover, because accuracy and distributional convergence increase in tandem, it may be the case that a cost function which more heavily penalizes distributional error (for example, hinge loss) may improve classification accuracy as well.

Moreover, it seems that the model does best with highly traded, liquid coins (such as ETH and LTC), indicating that our network may learn a returns structure driven by something akin to rational expectations, rather than by so-called "noise trader dynamics." Toward the end of the test set period, XRP experienced an unprecedented price run-up and subsequent correction. In such situations, it is well established that assessing asset value correctly during a bubble can cause short-term relative loss to an investor. XRP did not fully correct until late July 2017, so our test set results may simply reflect this fact. Due to returns correlation in all coins, this disparity may be reflected across returns.

The limited duration of our dataset creates some unavoidable bias due to sampling. Our model further displays high observed variance. We were not able to successfully remedy this with regularization, and we attribute this variance in part to the temporal partition of our dataset, which effects differences in the distribution of observations in our training, cross-validation, and test sets. Particularly in an emerging and rapidly changing asset market over a relatively limited duration, this distributional difference may create some degree of unavoidable variance.

6 Future Work

As discussed above, tuning algorithms with different loss functions may provide a ready opportunity for improvement of our model. We also limited the scope of our research to LSTM models and have left efficacy of GRU models to future investigation. There are several variations in model architecture still be explored as well - notably, the exploration of differential feature input times for autoregressive residuals and the proper feature set.

Beyond technical modifications such as this, we have discovered several exciting avenues for future work. Foremost, our results may point to limitations in our dataset (and publicly available data in general in the cryptocurrency space). A longer returns series would undoubtedly prove helpful for model training. Our results were also contextualized within high market volatility, and it may be the case that more granular (or coarser) timeframes in the return series could enhance predictive power of the model against this volatility. Furthermore, we believe that there is some predictability in volatility dynamics in the cryptocurrency market due to news events; incorporating news sentiment data or macroeconomic indicators may help our model to learn these dynamics as well. Finally, analysis of learned weights in the network may prove to reveal additional structure in the model predictions.

7 Contributions

We apportioned work equally for the project. Joel worked on developing the VAR and VARMAX Models. Ian developed the multi-coin RNN and R2N2 models and results analysis. Andrew worked on model development for the R2N2 and the class deliverables.

8 Codebase

`https : //github.com/ieshaw/deepcrypto.git`

References

[1] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(4), 2006.

[2] François Chollet and J.J. Allaire. Time series forecasting with recurrent neural networks. https://tensorflow.rstudio.com/blog/time-series-forecasting-with-recurrent-neural-networks.html. Accessed: 2018-1-28.

[3] E. Michael Azoff. *Neural Network Time Series Forecasting of Financial Markets*. John Wiley & Sons, Inc., 1994.

[4] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42, 2014.

[5] J. B. Heaton, N. G. Polson, and J. H. Witte. Deep learning in finance. 2017.

[6] Magnus Hansson. On stock return prediction with lstm networks, 2017.

[7] Kasun Bandara, Christoph Bergmeira, and Slawek Smylb. Forecasting across time series databases using long short-term memory networks on groups of similar series. 2017.

[8] Hardik Goel, Igor Melnyk, and Arindam Banerjee. R2n2: Residual recurrent neural networks for multivariate time series forecasting. 2017.

[9] G. Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[10] Rami Kawach. https://twitter.com/ramikawach/status/879405390118633472. Accessed: 2017-10-15.

[11] Statsmodels documentation: Varmax.