Hybrid Autoregressive-RNN Algorithm for Cryptocurrency Pricing
Joel Persson (joelpe@), Ian Shaw (iashaw@), Andrew Slottje (slottej@)
(1) Institute for Computational and Mathematical Engineering and (2) Management Science & Engineering, Stanford University

Objectives
- Discover arbitrage opportunities in cryptocurrency markets using exchange trading data to predict returns.
- Attenuate cryptocurrency market volatility through the use of this algorithm to correct mispricing.

Background
Predicting asset prices is a difficult problem because of the "no arbitrage principle," which states that markets price out excess returns. There is also limited previous research on cryptocurrency price prediction. Previous findings indicate suitability of LSTM models for this type of problem. [1, 2] One such model, the "RNN," uses vector autoregressions as RNN input. [3] We adapt this model to a multi-asset cryptocurrency setting and extend the model architecture to include residuals from a VARMAX regression on the feature set.

Data and Feature Selection
We use data published by the founder of Bitwren, an online cryptocurrency exchange. The data comprise hourly observations of various trading data as well as BTC-denominated price series in several coins. We select the top 5 by capitalization at initial date.
- Features: Volume, base volume, spread in selected coins, with vector autoregression residuals as additional features.
- Response: Log hourly returns in 5 coins.
- Data splitting: Training set over 9/2015 - 12/2016 (70%) with a cross-validation set over 12/2016 - 5/2017 (30%).
- Standardization: We use StandardScaler() from the sklearn library to normalize and center the feature set.

VARMAX Model Development
Vector autoregressive methods are used for multivariate time series problems. In the REN2 algorithm, residuals from the VAR prediction are used as inputs to an RNN. [4] We expand this method to include an adaptation of the more flexible and dynamic VARMAX model, which allows for moving average terms and error correction. [4]

\[ y_t = \alpha + \beta_t y_{t-1} + \ldots + \beta_t y_{t-k} + \epsilon_t + \sum_{i=1}^{p} \Delta_i y_{t-i} + \epsilon_t \]

We specify our feature set as the log returns of each coin and the 5 coin returns as endogenous \( y_t \in \mathbb{R}^5 \). We provide massive dimensionality reduction in the autoregressive optimization problem. This important computational improvement in REN2 time-series prediction is suitable for use in high-dimensional concrete spaces. We find the moving average terms are non-stationary and significant. Following our VARM analysis of AR(2) selected optimal lag length, we use a lag of 2 to specify the model.

RNN Model Development
Strong results in the literature indicate use of an LSTM network for this problem. LSTM cells are able to store values through many backpropagation epochs, thereby providing superior results for time-series data.
- Architecture search: Random dropping size architecture search selects 6 layers to be optimal, and random activation search selects a mix of tanh and ReLU. These results are robust to problem specification.
- Hyperparameter search: Small values for dropout and regularization hyperparameters work best, as shown at right; for Litecoin, batch sizes smaller than 72 were unstable under Adam optimization. EUC required additional regularization.
- Loss function: We train on balanced class weights. We fit mean-squared error loss for training predictions and cross-entropy loss against binary loss for prediction classification.
- Transfer learning: We iterate from a single-coin model to find regularization parameters and from the VARM to give the LSTM duration.

Results
Our model improves prediction accuracy by as much as 20% on both classes across coins, while our trading strategy is subject to volatility that diminishes returns. We plot the ROC and receiver operating curves (ROC) on the cross-validation and test sets. Although we obtain very good AUC and accuracy on the cross-validation set, these results do not fully transfer to the test set. While the loss functions decrease in random, indicating appropriate regularization, some variance is unavoidable as we have sampled from different durations to avoid market regime bias; the cross-validation set comes from a distribution which is closer in time to the training set than the test set.

Discussion
By our correlation metrics, we have shown that it is possible to build a strategy which generates excess returns in a portfolio of cryptocurrencies. This demonstrates efficiencies in the cryptocurrency market, in line with the "no arbitrage principle" which states that no such excess returns exist otherwise. It appears that high volatility and low liquidity negatively affect our classification accuracy and trading strategy results, in contrast to our expectation that our algorithm could exploit structure in currency volatility to boost accuracy. In particular, our results expose complex interplay between mean-reversion effects from the previous period’s returns and momentum effects that we hypothesize predominate under certain information regimes, which we believe to indicate observed currency volatility under information asymmetry. We accordingly support the following directions for future work:
- Adjusting investment frequency: There is a clear link between prediction accuracy and currency volatility.
- News data: Incorporating news sentiment data into the RNN implementation may yield resilience to volatility.
- Economic data: Including macroeconomic features in the VARMAX model may boost the model’s ability to learn seasonality.

References