A Time Series Approach to Explainability for Neural Nets with Applications to Risk-Management (and Fraud Detection)

Marc Wildi: ZHAW, marc.wildi@zhaw.ch
Branka Hadji Misheva: BFH, branka.hadjimisheva@bfh.ch

May 16, 2023

Table of contents

Neural Nets: Forecasting

- Review of **international forecast competitions**
- Before 2015: **classic linear** approaches **win**
  - M1 (1982), M2 (1993), M3 (2000), NN3 (2007) and NN5 (2009) competitions
  - We won the latter two with a 'sophisticated' linear approach
- Recently: **hybrid approaches** based on a mix of ARIMA and neural nets (NN) **outperform**
  - M4 (2020) and M5 (2021)
- Accruing interest in **neural nets for forecasting**: in particular for **economic** time series
- BUT... **Black Box**

XAI: Explainability

- Need a tool for **understanding/explaining** NN to trust model-output (forecasts)
- Classic XAI (eXplainable Artificial Intelligence)-approaches **not appropriate** for (autocorrelated) data
- Wildi-Hadji (2023): **time series XAI**-tool for **NN**

Regression and Explainability

- Let

$$y_t = 1 + 0.5x_{1,t} + 1.4x_{2,t} + \epsilon_t$$

- What let's you think you can understand/interpret this model?
- **Interpretation**: model-parameters are also **partial derivatives**
- A partial derivative is a concept that one can comprehend: story-telling
- **Wanted**: partial derivatives for NN
- **XAI**-tool: **first** and **second**-order partial derivatives of NN-output

NN: Partial Derivatives

- Let o_t be the output of a net fitted to x_{it}, $i = 1, ..., n$ (input/explanatory) and y_t (target), $t = 1, ..., T$
- Consider **partial derivatives** $\partial o_t / \partial x_{it}$ of NN-output o_t at each time point $t = 1, ..., T$ with respect to all input/explanatory variables $x_{1t}, ..., x_{nt}$
- Let $w_{it} := \partial o_t / \partial x_{it}$ and $b_t := o_t - \sum_{i=1}^n w_{it} x_{it}$
- Then $y_t = b_t + \sum_{i=1}^n w_{it} x_{it} + \epsilon_t$, $t = 1, ..., T$
- Wildi-Hadji (2023): **NN** as interpretable **time-dependent regression**

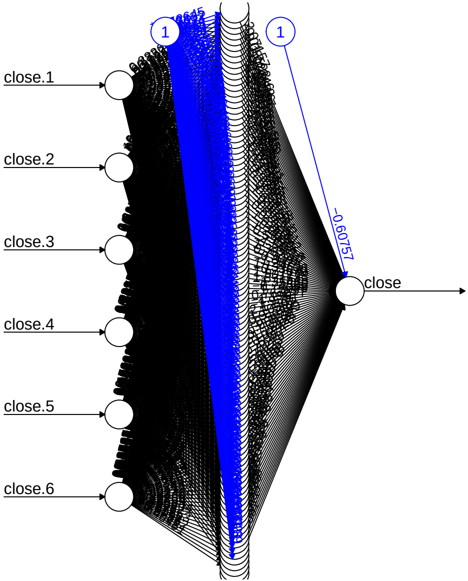
LPD and QPD

- Let $\mathbf{LPD}_t := (b_t, w_{1t}, ..., w_{nt})$ collect all 'regression' parameters (LPD:=Linear Parameter Data) ordered according to time
  - LPD computation: **exact, fast** and preserves **data-integrity** see Wildi-Hadji (2023)
- Let $\mathbf{QPD}_t := \partial\mathbf{LPD}_t/\partial\mathbf{x}_t$ be the **second-order** partial derivatives of o_t (diagonal elements of Hessian): Wildi-Hadji (2023)
  - $\mathbf{QPD}_t$ is a measure for **non-linearity** of NN and , by extension, non-linearity or 'complexity' of **market**

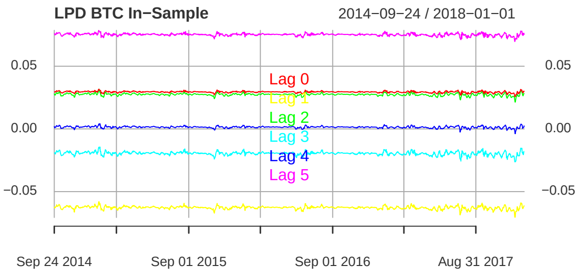
Application to Bitcoin (BTC)

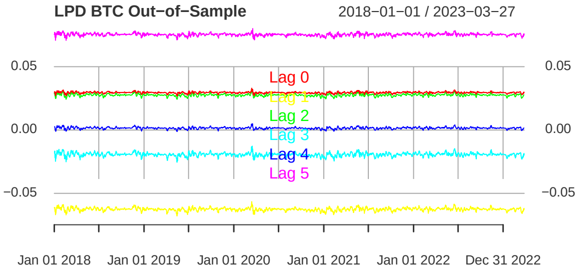
- Let r_t designate **log-returns** of bitcoin (BTC)
- Let $y_t := r_{t+1}$ be the future target and $x_{1t} = r_t, x_{2t} = r_{t-1}, ..., x_{6t} = r_{t-5}$ be contemporaneous and lagged explanatory or input variables
- Fit a NN to the data: **feedforward** with a single hidden layer of 100 neurons
  - Why feedforward? Can control **memory** of net
  - Why 100 neurons? Good compromise of **flexible** and **fast** (computing)
  - Results mostly robust across assets and architectures

NN: Feedforward Architecture



LPD In-Sample and Out-of-Sample





Outcome: Explainability for NN Applied to BTC

- Pretty much 'flat' LPD
- Mean value (mean over time) of LPD is

$$LPD = (0.03, -0.063, 0.027, -0.019, 0.001, 0.075)$$

- Compare with a OLS-**linear regression** model

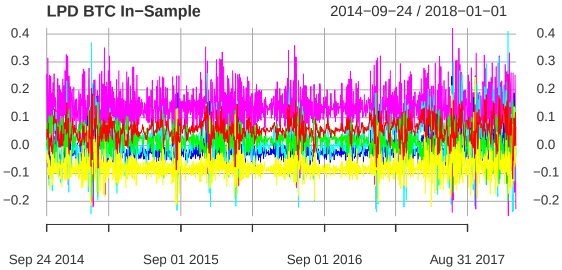
$$\begin{aligned} r_{t+1} &= 0.0002 + 0.03r_t - 0.063r_{t-1} + 0.027r_{t-2} - 0.019r_{t-3} \\ &\quad +0.001r_{t-4} + 0.075r_{t-5} \end{aligned}$$

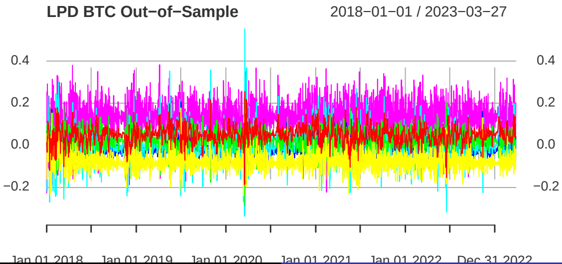
- **Remarkably** (!!!) both models match
  - Remarkable because net relies on 100*6+100 weights and 100+1 biases i.e. **801** parameters)

Outcome: Explainability for NN Applied to BTC

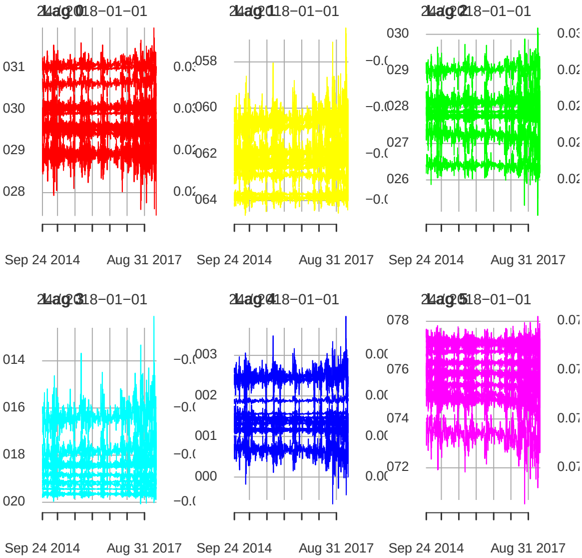
- XAI-outcome: in this example, one could substitute classic linear **regression** for (much more complex) non-linear **NN**
- Above net was fitted based on a **particular** random initialization (of 801 net parameters)
- What happens if we **change** the initial values? Effect of **numerical optimization**
  - Note: the overall effect does not depend on numerical optimization and/or R-packages and/or net architectures

LPD: Same as Above but Different Initialization of Net Parameters

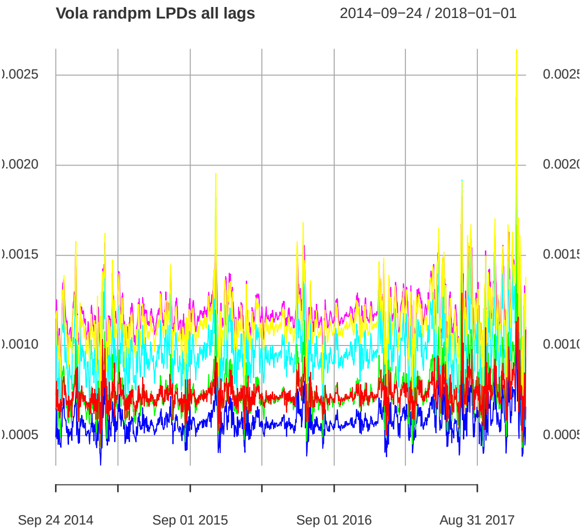




Ten Random LPDs Based on Ten Initializations



Volatility of Random LPDs



Outcomes

- **Explainablity**: replicate NN **exactly** by time-dependent regression (**LPD**)
- **Randomness**: interesting topic (but would more time)

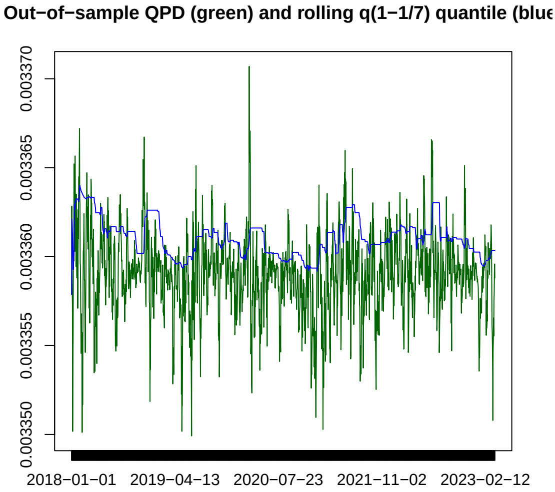
Explainability: Infer from NN to Market

- **Idea**: LPD is explanation for NN and, by extension, for data(-generating process) and **market**
- XAI-tool: real-time **market monitoring** (seismograph)
  - **Dependence structure** (of market): track LPD
  - **Non-linearity/non-stationarity** (of market): track QPD (second-order derivatives)
  - **Uncertainty** (of market): track volatility of random LPDs
- **Immediacy**: reduced latency (unlike classic momentum/MA-filters)

RM: Concept, Strategies and Operationalization

- **Risk-management** (RM-): we do not rely on **NN-output** o_t (forecasts), but on partial derivatives **LPD,QPD** of o_t
- **Concept**: downsize market-exposure (exit) when LPD or QPD are **'different than usual'**
- **Operationalization**: 4 strategies
  1. LPDs: above or below historical upper/lower quantiles (strong/weak market-**dependence**
  2. Volatility of random LPDs: above upper historical quantile (market-**inconclusiveness/uncertainty**)
  3. QPD: above historical upper quantile (market **non-linearity**)
  4. Weaker/smaller than usual **drift** (weak market-growth)
- Illustrate strategy 3: non-linearity, see Wildi-Hadji (2023)

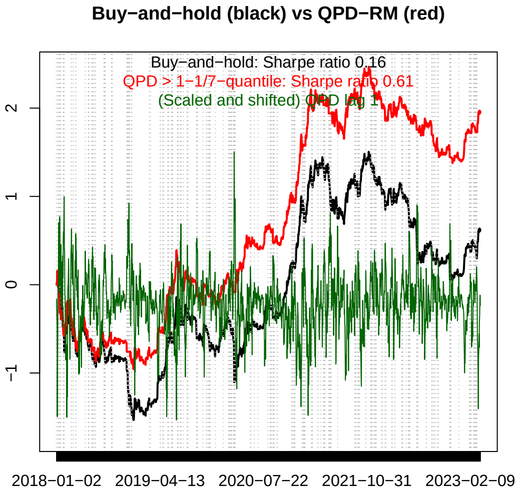
QPD (Non-Linearity: green) and Historical Quantile (blue)



Explanation

- QPD (dark green) is partial derivative of LPD: measure of local **non-linearity**
- Large QPD: dependence-structure non-linear (more complex data-structure)
- **Unusually** large QPD: if QPD exceeds historic rolling **quantile** (blue line)
- **Risk-Management** (RM-) strategy: **downsize** market exposure when QPD **exceeds** quantile (exit)

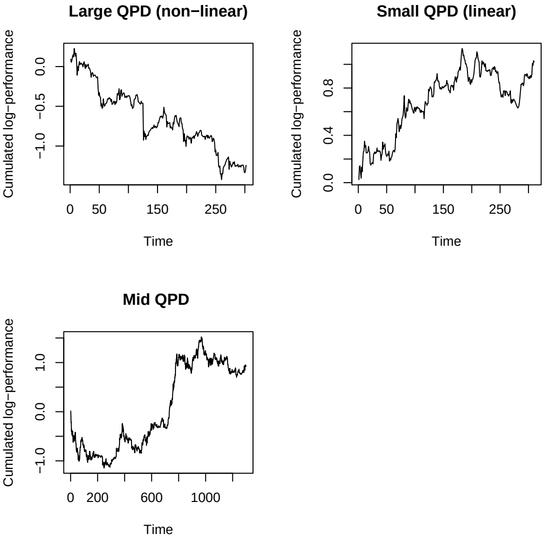
QPD-Based RM Out-Of-Sample: BTC (black), QPD (green), Exit when QPD Large (red)



Explanation

- Series are arbitrarily scaled (out-of-sample span starts in 2018)
- Buy-and-hold (black) vs. active RM-strategy (red)
  - QPD: green line
  - Shaded areas: QPD large (above rolling quantile on previous slide)
  - RM-strategy (red): ignore next day's BTC-return when today's QPD is large
- Large QPD (shaded areas) match some severe draw-downs of BTC

Performances : Critical, Neutral, Auspicious Time Points



Explanation

- Cumulated performances of (tomorrow's) BTC separated into episodes conditional on:
  - today's QPD 'large' (top left plot). Market non-linear!
  - today's QPD 'normal (between lower and upper quantile: top right plot). Market 'as usual'!
  - today's QPD 'small' (below lower quantile: bottom plot). Market 'linear'!
- **Magnitude** and **sign** of drift in the figures match **expectations**: see next slide for a **quantification** of effect

Performances : Critical, Neutral, Auspicious Time Points

	Prop. Positive sign	Average return
Critical time points	53%	-0.411%
Neutral time points	51%	0.072%
Auspicious time points	52.3%	0.329%
All time points	51.5%	0.037%

Table: Proportions of positive signs of next day's BTC-return (first column) and average next days' return (second column) based on critical time points (|QPD| > upper quantile: strong non-linearity), neutral time points (QPD in between: medium non-linearity), auspicious time points (|QPD| smaller than lower quantile: weak non-linearity) and all time points.

- QPD is inconclusive about 'signs'
- QPD seems to support prospective content about skewness/asymmetry/tail-risk

Summary

- We propose a novel **XAI**-tool for sensing the time dependent **reactiveness** of NN (partial derivatives: LPD and QPD)
- **Key inference**: from NN to (state of the) market
- **RM**: exit market when XAI-tool indicates 'abnormal' times
- **Small** latency
- We propose **four** different strategies based on different outputs of our XAI-tool (mixing)
- Link to Code on Github: R-code R-finance Chicago 2023