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

Marc Wildi and Branka Hadji Misheva

September 17, 2024

Neural Nets: Forecasting

1. AI is revolutionizing the world of analytics.

Neural Nets: Forecasting

1. AI is revolutionizing the world of analytics.
2. Forecast performances: review of **international Makridakis forecast competitions.**

Neural Nets: Forecasting

1. AI is revolutionizing the world of analytics.
2. Forecast performances: review of **international Makridakis forecast competitions.**
    - M1 (1982), M2 (1993), M3 (2000), NN3 (2007) and NN5 (2009) competitions: **classic linear** approaches **outperform** computationally intensive approaches (including neural nets).

Neural Nets: Forecasting

1. AI is revolutionizing the world of analytics.
2. Forecast performances: review of **international Makridakis forecast competitions.**
   - M1 (1982), M2 (1993), M3 (2000), NN3 (2007) and NN5 (2009) competitions: **classic linear** approaches **outperform** computationally intensive approaches (including neural nets).
   - M4 (2020) and M5 (2021): **hybrid approaches** based on a mix of ARIMA and neural nets **outperform** (hybrid approach won M4 competition).

Neural Nets: Forecasting

1. AI is revolutionizing the world of analytics.
2. Forecast performances: review of **international Makridakis forecast competitions.**
   - M1 (1982), M2 (1993), M3 (2000), NN3 (2007) and NN5 (2009) competitions: **classic linear** approaches **outperform** computationally intensive approaches (including neural nets).
   - M4 (2020) and M5 (2021): **hybrid approaches** based on a mix of ARIMA and neural nets **outperform** (hybrid approach won M4 competition).
3. Accruing interest in **neural nets for forecasting** (in particular economic time series).

Neural Nets: Main Problems/Issues

1. Problem 1: **random nets**, see paper (numerical optimization does not converge to the global optimum; different seeds, different estimates).

Neural Nets: Main Problems/Issues

1. Problem 1: **random nets**, see paper (numerical optimization does not converge to the global optimum; different seeds, different estimates).
2. Problem 2: **Black-box** or how to relate 'output' to 'input'? **Trust**?

Neural Nets: Main Problems/Issues

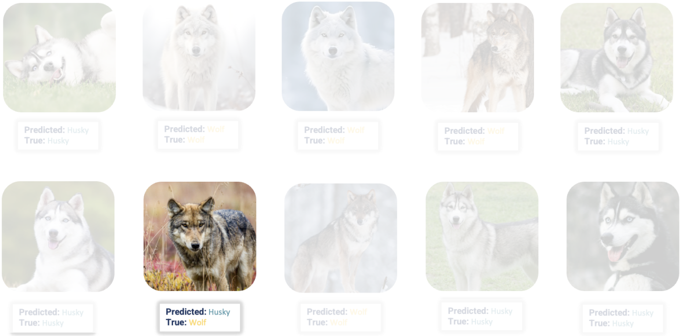
1. Problem 1: **random nets**, see paper (numerical optimization does not converge to the global optimum; different seeds, different estimates).
2. Problem 2: **Black-box** or how to relate 'output' to 'input'? **Trust**?
3. Problem 3: **Overfitting** of richly parameterized nets.

Why Do We Need Explainability?

Let's consider the 'Husky vs Wolf' experiment results.

Why Do We Need Explainability?

Let's consider the 'Husky vs Wolf' experiment results.



Why Do We Need Explainability?

Next, we investigate which features drive the classification.

Why Do We Need Explainability?

- Verify that **accuracy** is the result of **proper problem representation**.

Why Do We Need Explainability?

- Verify that **accuracy** is the result of **proper problem representation**.
  - The model is capturing **relevant dependencies** between features.

Why Do We Need Explainability?

- Verify that **accuracy** is the result of **proper problem representation**.
  - The model is capturing **relevant dependencies** between features.
  - This ensures **trust** in the system.

Why Do We Need Explainability?

- Verify that **accuracy** is the result of **proper problem representation**.
  - The model is capturing **relevant dependencies** between features.
  - This ensures **trust** in the system.
- **Communication**: convince layperson.

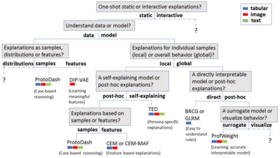
Why Do We Need Explainability?

- Verify that **accuracy** is the result of **proper problem representation**.
  - The model is capturing **relevant dependencies** between features.
  - This ensures **trust** in the system.
- **Communication**: convince layperson.
- **Regulation** demands it.

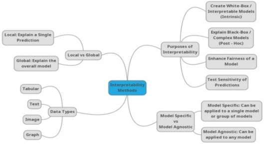
No black box excuses – explainability/traceability of models is necessary and can improve the analysis process | It is the responsibility of supervised firms to ensure that BDAI-based decisions can be explained and are understood by third-party experts. Supervisory authorities take a critical view of models that are categorised purely as black boxes. New approaches allow firms using such models to at least gain some insight into how these models work and identify the reasons behind decisions. In addition, a better understanding of models provides an opportunity to improve the analysis process – allowing, for instance, the responsible units in the supervised firm to identify statistical problems.

Figure: **Extract: Bafin AI and Big Data Report 2020**

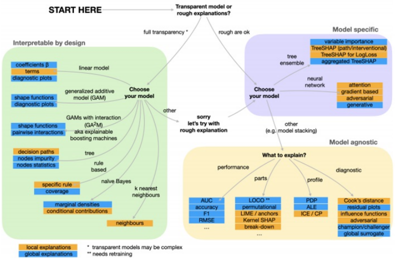
Deploying Explainability: Zoo of XAI models



Arya et al. (2019) proposed taxonomy based on questions about what is explained, how it is explained and at what level



Linardatos et al. (2021) taxonomy mind-map of Machine Learning Interpretability Techniques.



Maksymiuk et al. (2021) model-oriented taxonomy for XAI method

Figure: **Machine Learning Interpretability Techniques**

LIME - I

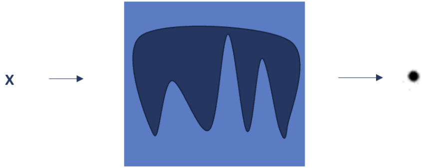
- LIME stands for **Local, Interpretable, Model-agnostic Explanations.**

LIME - I

- LIME stands for **Local, Interpretable, Model-agnostic Explanations.**
- It explains the prediction of any machine learning classifier by learning an interpretable model locally around the prediction.

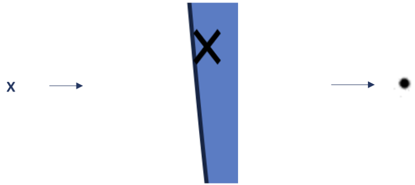
LIME - I

- LIME stands for **Local, Interpretable, Model-agnostic Explanations.**
- It explains the prediction of any machine learning classifier by learning an interpretable model locally around the prediction.



LIME - II

- LIME stands for **Local, Interpretable, Model-agnostic Explanations.**
- It explains the prediction of any machine learning classifier by learning an interpretable model locally around the prediction.



LIME - III

- Formally, the explanation provided by LIME for each observation:

$$\xi(x) = \arg\min_{g \in G} \; L(f, g, \pi_x) + \Omega(g) \tag{1}$$

where, f is the original model,
$g \in G$ is the interpretable model,
$L(f, g, \pi_x)$ is the loss function,
π_x is the proximity measure around x,
$\Omega(g)$ is the complexity of g.

- The goal is to minimize the locality-aware loss L without making any assumptions about f, since a key property of LIME is that it is model-agnostic.

SHAP Values

- Given a model

$$f(x_1, x_2, x_3, \ldots, x_n) \tag{2}$$

- Marginal contribution $\Delta_v(i, S)$ of a feature i:

$$\Delta_v(i, S) = v(S \cup \{i\}) - v(S) \tag{3}$$

- Let Π be the set of permutations of the integers up to N, and given $\pi \in \Pi$ let

$$S_{i,\pi} = \{j : \pi(j) < \pi(i)\} \tag{4}$$

be the players preceding player i in π, then:

$$\phi_v(i) = \frac{1}{N!} \sum_{\pi \in \Pi} \Delta_v(i, S_{i,\pi}) \tag{5}$$

- Shapley value is the weighted average of the feature's marginal contribution to every possible subset of features.

Deploying Explainability: Utility of Classical XAI Methods for Finance

- Classic approaches are **data-intensive**.
- Solution:
  - Create '**new**' data (simulation), or
  - **Reshuffle** available data.
- Problems:
  - if features are correlated, the artificial coalitions created will lie outside of the multivariate joint distribution of the data.
  - if the data are independent, coalitions can still be meaningless.
  - perturbation-based methods are fully dependent on the ability to perturb samples in a meaningful way which is not always the case with financial data (ex. one-hot encoding).

Example: BTC Time Series (Time-Ordered Data)

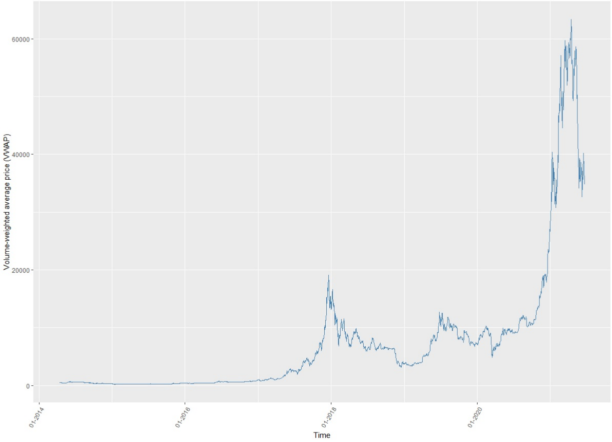


Figure: **BTC Prices (correct ordering)**

Example: Re-Shuffling BTC

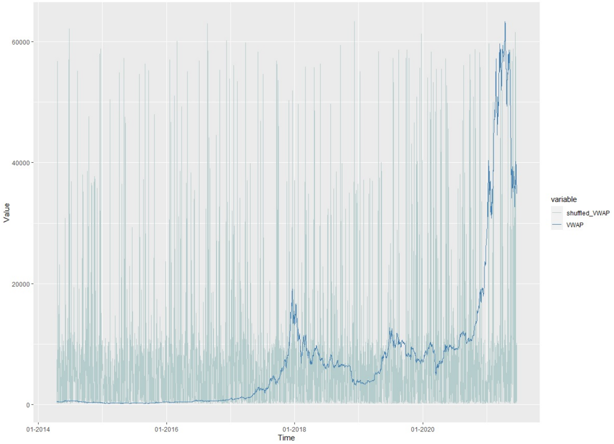


Figure: **BTC Prices (correct ordering vs shuffled values)**

Explainability Example 1: Classic Regression

- Let

$$y_t = 1 + 0.5x_{1,t} + 1.4x_{2,t} + \epsilon_t \qquad (6)$$

Explainability Example 1: Classic Regression

- Let

$$y_t = 1 + 0.5x_{1,t} + 1.4x_{2,t} + \epsilon_t \tag{6}$$

- **Interpretation**: what is the meaning of the parameters?

Explainability Example 1: Classic Regression

- Let

$$y_t = 1 + 0.5x_{1,t} + 1.4x_{2,t} + \epsilon_t \qquad (6)$$

- **Interpretation**: what is the meaning of the parameters?
- **Communication**: regression is 'well-known'

Explainability Example 1: Classic Regression

- Let

$$y_t = 1 + 0.5x_{1,t} + 1.4x_{2,t} + \epsilon_t \quad (6)$$

- **Interpretation**: what is the meaning of the parameters?
- **Communication**: regression is 'well-known'
- **Utility**: change $x_{1,t}, x_{2,t}$ in view of achieving a certain target y_t (macro-economic model); **policy**

Explainability Example 1: Classic Regression

- Let

$$y_t = 1 + 0.5x_{1,t} + 1.4x_{2,t} + \epsilon_t \quad (6)$$

- **Interpretation**: what is the meaning of the parameters?
- **Communication**: regression is 'well-known'
- **Utility**: change $x_{1,t}, x_{2,t}$ in view of achieving a certain target y_t (macro-economic model); **policy**
- **Validation**: confrontation with common sense/expert knowledge/**experience**

Explainability Example 2: Time Series

- Consider two simple **forecast** rules

$$\hat{x}_{t+1} = 0.2x_t + 0.2x_{t-1} + 0.2x_{t-2} + 0.2x_{t-3} + 0.2x_{t-4}$$
$$\hat{x}_{t+1} = 0.5x_t + 0.25x_{t-1} + 0.13x_{t-2} + 0.06x_{t-3} + 0.03x_{t-4}$$

Explainability Example 2: Time Series

- Consider two simple **forecast** rules

$$\hat{x}_{t+1} = 0.2x_t + 0.2x_{t-1} + 0.2x_{t-2} + 0.2x_{t-3} + 0.2x_{t-4}$$
$$\hat{x}_{t+1} = 0.5x_t + 0.25x_{t-1} + 0.13x_{t-2} + 0.06x_{t-3} + 0.03x_{t-4}$$

- **Interpretation**
  - Remote past is as important as present time
  - Remote past is less relevant than recent data

Explainability Example 2: Time Series

- Consider two simple **forecast** rules

$$
\begin{aligned}
\hat{x}_{t+1} &= 0.2x_t + 0.2x_{t-1} + 0.2x_{t-2} + 0.2x_{t-3} + 0.2x_{t-4} \\
\hat{x}_{t+1} &= 0.5x_t + 0.25x_{t-1} + 0.13x_{t-2} + 0.06x_{t-3} + 0.03x_{t-4}
\end{aligned}
$$

- **Interpretation**
  - Remote past is as important as present time
  - Remote past is less relevant than recent data
- **Learn** from model: dynamics of time series
- **Story-telling**

New XAI-Tool

- **Preserve dependence** (no re-shuffling).

New XAI-Tool

- **Preserve dependence** (no re-shuffling).
- Avoid **inventing** (no simulation).

New XAI-Tool

- **Preserve dependence** (no re-shuffling).
- Avoid **inventing** (no simulation).
- Link to **regression** (no exoticism).

X-Function - I

- We propose a family of **explainability (X-)functions** $xf(\cdot)$ for assigning meaning to the net's response or output $\mathbf{o_t}$ over time $t = 1, ..., T$, where $\mathbf{o_t} = (o_{1t}, ..., o_{n_pt})$ is a n_p dimensional vector of possibly multiple output neurons.

X-Function - I

- We propose a family of **explainability (X-)functions** $xf(\cdot)$ for assigning meaning to the net's response or output $\mathbf{o_t}$ over time $t = 1, ..., T$, where $\mathbf{o_t} = (o_{1t}, ..., o_{n_pt})$ is a n_p dimensional vector of possibly multiple output neurons.
- By selecting the identity $xf(\mathbf{o_t}) = \mathbf{o_t}$, we can mark preference for the sensitivities or partial derivatives $w_{ijt} := \partial o_{jt}/\partial x_{it}$, $i = 1, ..., n$, $j = 1, ..., n_p$, for each explanatory variable x_{it} of the net.

X-Function - I

- We propose a family of **explainability (X-)functions** $xf(\cdot)$ for assigning meaning to the net's response or output $\mathbf{o_t}$ over time $t = 1, ..., T$, where $\mathbf{o_t} = (o_{1t}, ..., o_{n_p t})$ is a n_p dimensional vector of possibly multiple output neurons.
- By selecting the identity $xf(\mathbf{o_t}) = \mathbf{o_t}$, we can mark preference for the sensitivities or partial derivatives $w_{ijt} := \partial o_{jt}/\partial x_{it}$, $i = 1, ..., n$, $j = 1, ..., n_p$, for each explanatory variable x_{it} of the net.
- In order to complete the 'explanation', we can add a synthetic intercept to each output neuron o_{jt} defined according to:

$$b_{jt} := o_{jt} - \sum_{i=1}^{n} w_{ijt} x_{it} \tag{7}$$

X-Function - II

- For each output neuron o_{jt}, the resulting derivatives or 'explanations' $b_{jt}, w_{1jt}, ..., w_{njt}$ generate a new data-flow which is referred to as **Linear Parameter Data (LPD)**.

X-Function - II

- For each output neuron o_{jt}, the resulting derivatives or 'explanations' $b_{jt}, w_{1jt}, ..., w_{njt}$ generate a new data-flow which is referred to as **Linear Parameter Data (LPD)**.
- The **LPD** is a matrix of dimension $T * (n+1)$, irrespective of the complexity of the neural net, with $t-$th row denoted by $\mathbf{LPD}_{jt} := (b_{jt}, w_{1jt}, ..., w_{njt})$.

X-Function - II

- For each output neuron o_{jt}, the resulting derivatives or 'explanations' $b_{jt}, w_{1jt}, ..., w_{njt}$ generate a new data-flow which is referred to as **Linear Parameter Data (LPD)**.
- The **LPD** is a matrix of dimension $T * (n+1)$, irrespective of the complexity of the neural net, with $t-$th row denoted by $\mathbf{LPD}_{jt} := (b_{jt}, w_{1jt}, ..., w_{njt})$.
- The **LPD** can be interpreted in terms of exact replication of the net by a linear model at each time point t and the *natural* time-ordering of $\mathbf{LPD}_{jt}$ subsequently allows to examine changes of the linear replication as a function of time.

X-Function - II

- For each output neuron o_{jt}, the resulting derivatives or 'explanations' $b_{jt}, w_{1jt}, ..., w_{njt}$ generate a new data-flow which is referred to as **Linear Parameter Data (LPD)**.
- The **LPD** is a matrix of dimension $T * (n+1)$, irrespective of the complexity of the neural net, with $t-$th row denoted by $\mathbf{LPD}_{jt} := (b_{jt}, w_{1jt}, ..., w_{njt})$.
- The **LPD** can be interpreted in terms of exact replication of the net by a linear model at each time point t and the *natural* time-ordering of $\mathbf{LPD}_{jt}$ subsequently allows to examine changes of the linear replication as a function of time.
- We are then in a position to assign a meaning to the neural net, at each time point t = 1, ..., T, and to monitor non-linearities of the net or, by extension, possible non-stationarities of the data.

Toy Example: Time Series Forecasting

Model Structure:

- **Inputs**: 6 lagged values.
- **Output**: Predicted return at the next time step.
- **Activation function**: Sigmoid at the output layer.

Toy Example: Time Series Forecasting

Model Structure:

- **Inputs**: 6 lagged values.
- **Output**: Predicted return at the next time step.
- **Activation function**: Sigmoid at the output layer.

Model Equation:

$$y = \sigma(W_1 \cdot X_1 + W_2 \cdot X_2 + \ldots + W_6 \cdot X_6 + b) \tag{8}$$

- $W_1, W_2, \ldots, W_6$: Weights for each lagged input.
- b: Bias term.
- $\sigma(z) = \frac{1}{1+e^{-z}}$: Sigmoid activation function.

Toy Example: Time Series Forecasting

Weighted sum:

$$z = W_1 \cdot X_1 + W_2 \cdot X_2 + \ldots + W_6 \cdot X_6 + b \tag{9}$$

Toy Example: Time Series Forecasting

Weighted sum:

$$z = W_1 \cdot X_1 + W_2 \cdot X_2 + \ldots + W_6 \cdot X_6 + b \tag{9}$$

Apply the Sigmoid function:

$$y = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Toy Example: Time Series Forecasting

Weighted sum:

$$z = W_1 \cdot X_1 + W_2 \cdot X_2 + \ldots + W_6 \cdot X_6 + b \quad (9)$$

Apply the Sigmoid function:

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (10)$$

Derivative of the output with respect to X_i:

$$\frac{\partial y}{\partial X_i} = \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial X_i} \quad (11)$$

$$\frac{\partial y}{\partial z} = y(1 - y) \quad (12)$$

$$\frac{\partial y}{\partial X_i} = y(1 - y) \cdot W_i \quad (13)$$

Toy Example: Time Series Forecasting

Sensitivity interpretation:

- Indicates how sensitive the predicted return y is to changes in the lagged value X_i.
- If W_i is large, X_i significantly impacts the predicted return.
- If W_i is small, X_i has little impact on the prediction.

Toy Example: Time Series Forecasting

Sensitivity interpretation:

- Indicates how sensitive the predicted return y is to changes in the lagged value X_i.
- If W_i is large, X_i significantly impacts the predicted return.
- If W_i is small, X_i has little impact on the prediction.

Calculation:

$$W_1 = 0.5, \quad W_2 = -0.2, \quad W_3 = 0.1, \quad W_4 = 0.05$$

$$W_5 = -0.3, \quad W_6 = 0.7, \quad b = 0.1$$

Assume $z = 0.6$:

$$y = \sigma(0.6) \approx 0.645$$

$$\frac{\partial y}{\partial X_1} = 0.645(1 - 0.645) \cdot 0.5 \approx 0.114$$

$$\frac{\partial y}{\partial X_2} = 0.645(1 - 0.645) \cdot (-0.2) \approx -0.046$$

Context

- Derivation of LPD: **formula**, see paper
    - We derive a formal expression for the LPD.

Context

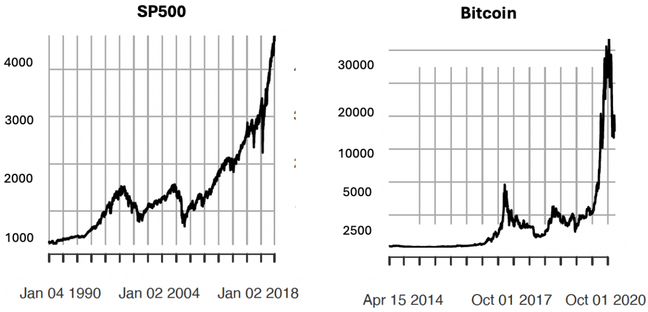
- Derivation of LPD: **formula**, see paper
  - We derive a formal expression for the LPD.
  - We also assume that all neurons have a Sigmoid activation function.

Context

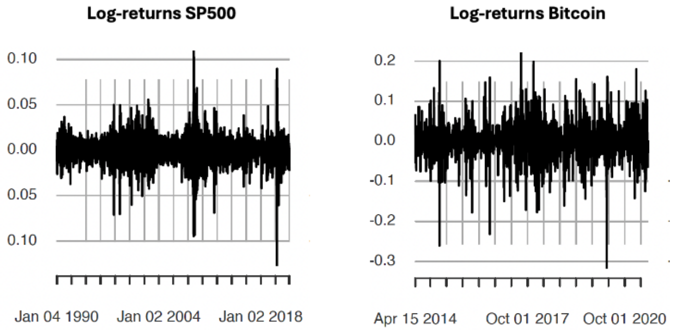
- Derivation of LPD: **formula**, see paper
  - We derive a formal expression for the LPD.
  - We also assume that all neurons have a Sigmoid activation function.
  - Straightforward modifications apply in the case of arbitrary differentiable activation functions.

Risk Management: Application of LPD

- We apply the LPD to the Bitcoin (BTC) and to the S&P500 equity index.



Risk Management: Application of LPD



Risk Management: Application of LPD to BTC Cryptocurrency

- **Data**: Bitcoin returns 15-04-2014 to 30-06-2021

Risk Management: Application of LPD to BTC Cryptocurrency

- **Data**: Bitcoin returns 15-04-2014 to 30-06-2021
- **Model**: simple feedforward net with a single hidden layer of dimension one hundred and an input layer collecting the last six lagged (daily) returns: the net is then trained to predict next day's return based on the MSE-criterion. The number of estimated parameters then amounts to a total of $6x100 + 100 = 700$ weights and $100 + 1 = 101$ biases.

$$B\hat{T}C_{T+1} = NN(BTC_T, BTC_{T-1}, ..., BTC_{T-5}) \tag{14}$$

Risk Management: Application of LPD to BTC Cryptocurrency

- **Data**: Bitcoin returns 15-04-2014 to 30-06-2021
- **Model**: simple feedforward net with a single hidden layer of dimension one hundred and an input layer collecting the last six lagged (daily) returns: the net is then trained to predict next day's return based on the MSE-criterion. The number of estimated parameters then amounts to a total of $6x100 + 100 = 700$ weights and $100 + 1 = 101$ biases.

$$B\hat{T}C_{T+1} = NN(BTC_T, BTC_{T-1}, ..., BTC_{T-5}) \qquad (14)$$

- We optimize the net 100-times, based on different random initializations of its parameters, and we compute trading performances of each random-net based on a simple sign-rule.

Performance

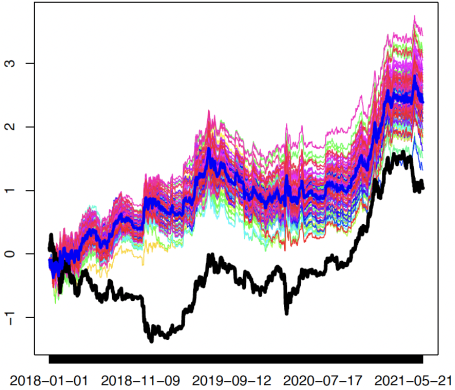


Figure: Cumulative log-performances out-of-sample based on a sign-rule (buy or sell depending on the sign of forecasted return) vs. buy-and-hold (bold black) and mean-net performance (bold blue)

Interpretability: LPD and Variable Relevance

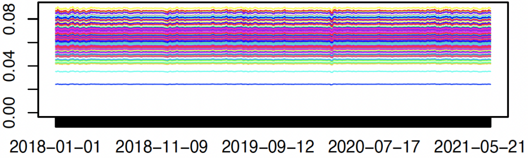


Figure: LPDs of 100 random nets for the 6th lagged return

Interpretability: LPD and Variable Relevance

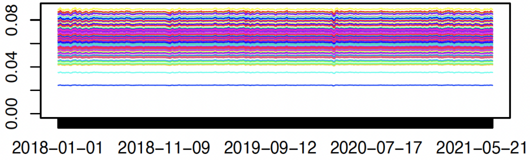


Figure: LPDs of 100 random nets for the 6th lagged return

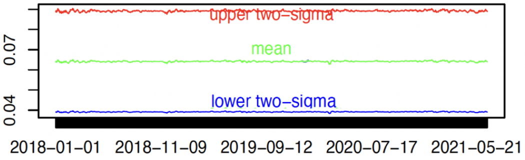


Figure: Mean and two sigma band of above LPD realizations

Interpretability: LPD and Variable Relevance

- Since the LPD corresponds to the parameters of a (time-dependent) linear replication of the net, synthetic t-statistics could be computed for inferring the relevance of the explanatory variables by computing the ratio of mean-LPD and standard-deviation:

$$\mathbf{t}_t := \frac{\overline{\mathbf{LPD}}_t}{\boldsymbol{\sigma}_t} \tag{15}$$

at each time point t, corresponding to (a vector of) synthetic t-statistics, one for each input variable.

LPDs: Additional Findings

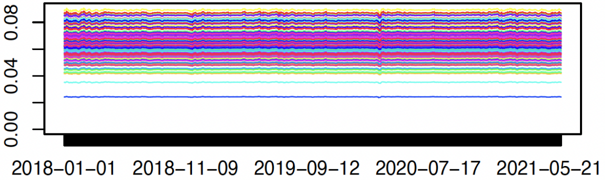


Figure: LPDs of 100 random nets for the 6th lagged return

LPDs: Additional Findings

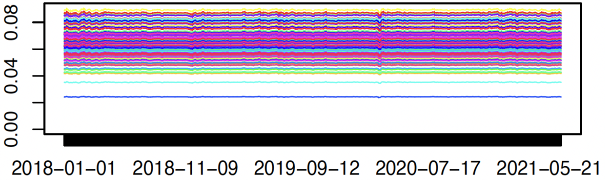


Figure: LPDs of 100 random nets for the 6th lagged return

- We observe nearly constant sensitivities along the time axis.

LPDs: Additional Findings

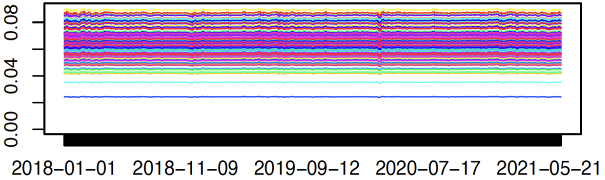


Figure: LPDs of 100 random nets for the 6th lagged return

- We observe nearly constant sensitivities along the time axis.

$$\overline{o}_t = \frac{1}{100}\sum_{i=1}^{100} o_{t,i} = \overline{\mathbf{LPD}}_t \begin{pmatrix} 1 \\ \mathbf{x}_t \end{pmatrix} \approx 0.0015 + 0.065 \sum_{j=1}^{6} x_{jt} \quad (16)$$

where $\overline{\mathbf{LPD}}_t$ is the vector of mean-intercept and mean-LPDs.

LPDs and Risk Management

- **Remember:** We estimate the LPDs for 100 random nets.
  - Next, we observe the dependency of the LPDs across the different random nets.

LPDs and Risk Management

- **Remember:** We estimate the LPDs for 100 random nets.
  - Next, we observe the dependency of the LPDs across the different random nets.
- **Idea:**
  - Identify **'uncertain times'** or **'unusual states'** of the market in real-time.
  - **Downsize** market exposure during **uncertain times**/unusual states.

LPDs and Risk Management

- **Remember:** We estimate the LPDs for 100 random nets.
  - Next, we observe the dependency of the LPDs across the different random nets.
- **Idea:**
  - Identify **'uncertain times'** or **'unusual states'** of the market in real-time.
  - **Downsize** market exposure during **uncertain times**/unusual states.
- **Findings:**
  - The time-varying dependency of the data measured by the LPDs is indicative of different states of the market.
  - In particular, weak dependency (small absolute LPD) is an indicator of randomness or chaos.
  - Therefore, we propose a simple rule for managing risks: exit markets at times tagged as chaotic by the LPD.

Market-Exit when |LPD| Weak (Critical Time)

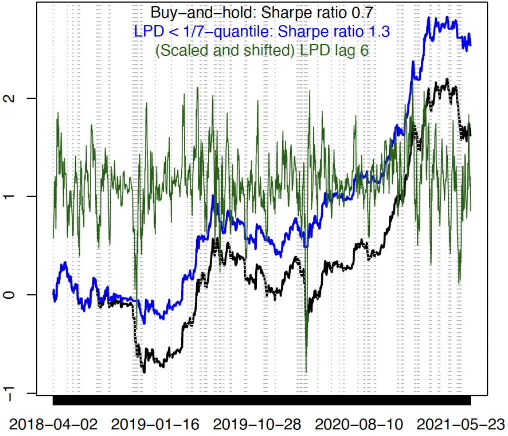


Figure: Buy-and-hold (black) vs. Out-of-sample (mean-) LPD market-exit strategy (blue): exits (shaded in grey) occur if today's out-of-sample mean-LPD (green) drops below the 1/7 quantile.

Summary

- Classic XAI methods are not suited for financial time series data.

Summary

- Classic XAI methods are not suited for financial time series data.
- **X-functions and tool developed**: exact, fast, intuitive and preserves the natural time ordering of data.

Summary

- Classic XAI methods are not suited for financial time series data.
- **X-functions and tool developed**: exact, fast, intuitive and preserves the natural time ordering of data.
- Can be used to explain variable importance though proper statistical testing and can be used for risk management.

Summary

- Classic XAI methods are not suited for financial time series data.
- **X-functions and tool developed**: exact, fast, intuitive and preserves the natural time ordering of data.
- Can be used to explain variable importance though proper statistical testing and can be used for risk management.
- Please refer to our paper and code:
  - Paper
  - Code