Poster ID: 5185

# Augmenting Transferred Representations for Stock Classification

Elizabeth Fons[1], Paula Dawson[2], Xiao-jun Zeng[1], John Keane[1], Alexandros Iosifidis[3]

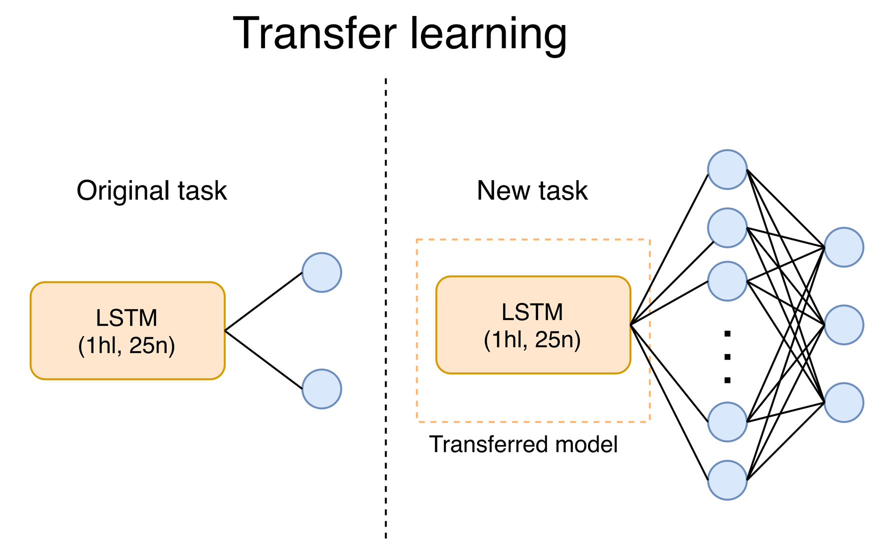
[1]University of Manchester, Manchester, UK. [2]AllianceBernstein, London, UK. [3]Aarhus University, Aarhus, Denmark.

## Introduction

Stock classification is a challenging task due to high levels of noise and volatility of stocks returns.

- We show that using transfer learning can help, by pre-training a model to extract universal features on the full universe of stocks of the S&P500 index and then transferring it to another model to directly learn a trading rule.
- We propose the use of data augmentation on the feature space defined as the output of a pre-trained model (i.e., augmenting the aggregated time-series representation) and compare this approach the standard augmentation in the input space.
- We test our model by building the learned trading rule and calculate profitability considering transaction fees.
- Transferred models present more than double the risk-adjusted returns than their counterparts and augmentation methods on the feature space leads to 20% increase in risk-adjusted return compared to a transferred model without augmentation.

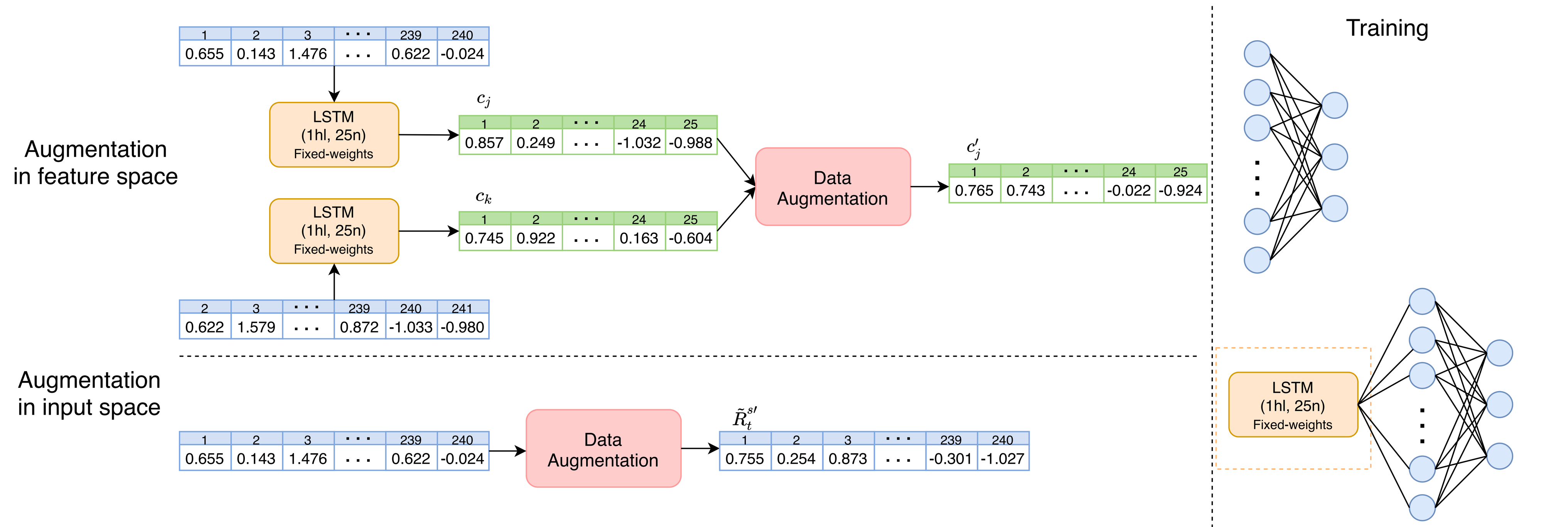
## Methodology



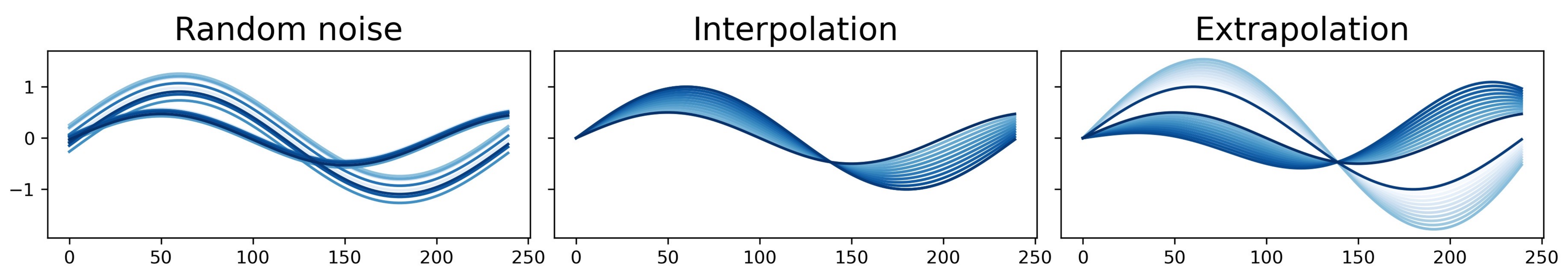

- Dataset: all constituent stocks of the S&P500 index, from1990 to 2018.
- Original task: binary classification, above daily median (trend up) or below (trend down). Network: single layer LSTM with 25n, and a fully connected two-neuron output.
- New task: trading rule with top $K$ stocks labelled *buy*, bottom $K$ stocks labelled *sell* and the rest as *do nothing*. Network: single layer LSTM (25n) fixed with a fully connected layer of {25,100} neurons and an output layer of 3.
- Loss: we used the cross-entropy loss and incorporated a loss term that optimizes the average return, as follows:

$$\mathcal{L}_{R+CE}(\Theta) = \mathcal{L}_{CE} + \alpha\mathcal{L}_{returns} = \mathcal{L}_{CE} + -\alpha\frac{1}{B}\sum R(i,t)$$

## Augmenting transferred representations




### Data augmentation in feature space:




- Interpolation [1]: for each sample, we find its $K$ intra-class nearest neighbors in feature space and for each pair a new vector is generated using:

$$\mathbf{c}'_j = (\mathbf{c}_k - \mathbf{c}_j)\lambda + \mathbf{c}_j$$

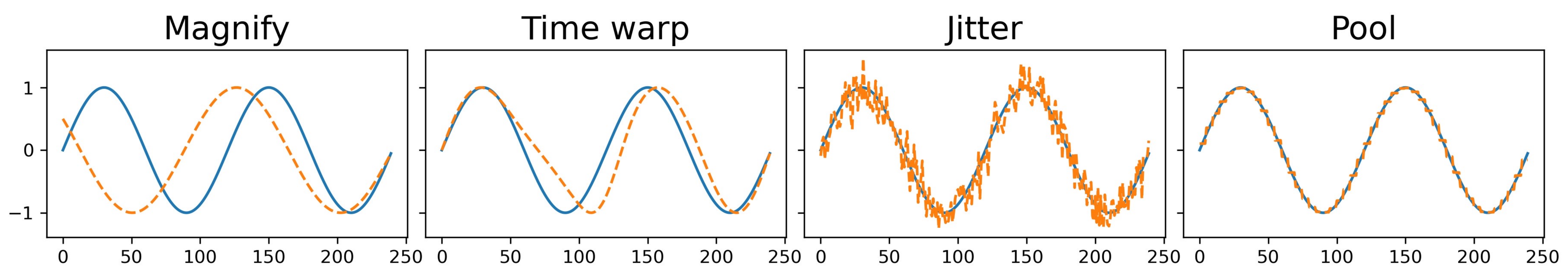
- Extrapolation: similarly, we apply extrapolation to the feature space vectors:

$$\mathbf{c}'_j = (\mathbf{c}_j - \mathbf{c}_k)\lambda + \mathbf{c}_j$$

- Noise: Gaussian noise is generated with zero mean and per-element standard deviation calculated across all transformed vectors in the dataset

$$\mathbf{c}'_i = \mathbf{c}_i + \gamma X, X \sim \mathcal{N}\{0, \sigma_i^2\}$$

### Data augmentation in input space:




- Magnify: a variation of window slicing [2], we randomly slice windows between 40% and 80% of the original time series, but always from the fixed end.
- Jitter: Gaussian noise with $\mu = 0$ and standard deviation $\sigma = 0.05$ is added [3].
- Pool: Reduces the temporal resolution without changing the length of the time series by averaging a pooling window.
- Time warp: time intervals between samples are distorted based on a random smooth warping curve by cubic spline with four knots at random magnitude [3].

## Results

- Ann ret and vol: annualized return and annualized volatility.
- Information Ratio (IR): ratio between excess return (portfolio returns minus benchmark returns) and tracking error (standard deviation of excess returns).
- Downside information ratio (DIR) ratio between excess return and the down-side risk (D. Risk: variability of under-performance below the benchmark), that differentiates harmful volatility from total overall volatility.

| Method | Ann ret | Ann vol | IR | D. Risk | DIR | Acc | Macro-F1 |
|---|---|---|---|---|---|---|---|
| LSTM [4] | 29.2 | **28.66** | 1.02 | 19.08 | 1.53 | — | — |
| No TL (25)+$\mathcal{L}_{CE}$ | 12.99 | 38.15 | 0.34 | 25.48 | 0.52 | 73.13±18.94 | **33.57 ± 6.5** |
| TL+FC(25)+$\mathcal{L}_{CE}$ | 32.25 | 30.29 | 1.06 | 19.6 | 1.65 | 68.34±16.5 | 31.79±5.12 |
| TL+FC(25)+$\mathcal{L}_{R+CE}$ | 34.62 | 30.20 | 1.15 | 19.59 | 1.77 | 64.79±16.86 | 30.72±5.28 |
| TL+FC(25) Extrapolation | **39.70** | 29.43 | **1.35** | 18.96 | **2.09** | 62.90±17.87 | 30.10±5.81 |
| TL+FC(25) Interpolation | 36.87 | 29.69 | 1.24 | **18.93** | 1.95 | 62.46±17.80 | 29.95±5.74 |
| TL+FC(25) Noise | 30.97 | 29.15 | 1.06 | 19.14 | 1.62 | 62.43±18.12 | 29.95±5.81 |
| TL+FC(25) Jitter-feat | 39.11 | 29.93 | 1.31 | 19.22 | 2.03 | 62.71±17.84 | 30.04±5.71 |
| TL+FC(25) Jitter-input | 29.74 | 39.94 | 0.96 | 20.12 | 1.48 | 68.23±16.62 | 31.75±5.06 |
| TL+FC(25) Magnify | 20.39 | 29.41 | 0.69 | 19.86 | 1.03 | 63.78±16.78 | 30.42±5.47 |
| TL+FC(25) Pool | 27.18 | 29.96 | 0.91 | 19.64 | 1.38 | 57.71±17.43 | 28.38±5.72 |
| TL+FC(25) Time Warp | 32.76 | 29.46 | 1.11 | 19.21 | 1.71 | 61.81±19.96 | 29.80±5.48 |

| Method | Ann ret | Ann vol | IR | D. Risk | DIR | Acc | Macro-F1 |
|---|---|---|---|---|---|---|---|
| LSTM [4] | 29.2 | 28.66 | 1.02 | 19.08 | 1.53 | — | — |
| No TL (100)+$\mathcal{L}_{R+CE}$ | 21.05 | 39.95 | 0.55 | 25.9 | 0.84 | 57.02±21.95 | 28.24±8.12 |
| TL+FC(100)+$\mathcal{L}_{CE}$ | 30.83 | 30.31 | 1.02 | 19.79 | 1.56 | 68.88±15.93 | 31.95±4.76 |
| TL+FC(100)+$\mathcal{L}_{R+CE}$ | 32.14 | 29.97 | 1.07 | 19.87 | 1.62 | 64.72±17.25 | 30.7±5.41 |
| TL+FC(100) Extrapolation | 27.38 | 29.33 | 0.93 | 19.42 | 1.41 | 62.74±17.73 | 30.05±5.81 |
| TL+FC(100) Interpolation | 30.84 | 29.72 | 1.04 | 19.38 | 1.59 | 62.49±17.35 | 30.01±5.63 |
| TL+FC(100) Noise | 29.02 | 29.44 | 0.99 | 19.2 | 1.51 | 62.2±17.82 | 29.87±5.73 |
| TL+FC(100) Jitter-feat | **37.14** | 29.31 | **1.27** | **18.92** | **1.96** | 61.84±17.89 | 29.75±5.75 |
| TL+FC(100) Jitter-input | 29.49 | 30.32 | 0.97 | 19.73 | 1.49 | 67.79±17.18 | 31.64±5.46 |
| TL+FC(100) Magnify | 22.11 | 30.36 | 0.73 | 20.21 | 1.09 | 67.12±16.68 | 30.34±5.27 |
| TL+FC(100) Pool | 27.64 | 29.50 | 0.94 | 18.98 | 1.46 | 57.64±17.89 | 28.37±5.89 |
| TL+FC(100) Time Warp | 26.64 | 29.65 | 0.90 | 19.49 | 1.37 | 65.55±18.03 | 29.69±5.89 |

## References

[1] Terrance DeVries and Graham W. Taylor, "Dataset augmentation in feature space", 2017.

[2] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard, "Data augmentation for time series classification using convolutional neural networks", in *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2016.

[3] Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulic, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks", in *Proceedings of the 19th ACM International Conference on Multimodal Interaction, 2017, ICMI '17*, p. 216–220.

[4] Thomas Fischer and Christopher Krauss, "Deep learning with long short-term memory networks for financial market predictions", *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.