Supplementary Materials: Forecasting Financial Market Structure from Network Features using Machine Learning

Douglas Castilho[1,5,*], Thársis T. P. Souza[2], Soong Moon Kang[3], João Gama[4], André C. P. L. F. de Carvalho[1]

[1]Institute of Mathematics and Computer Sciences (ICMC),
University of São Paulo (USP), São Carlos, Brazil,
[2]Department of Computer Science, University College London,
Gower Street, London, WC1E 6BT, UK
[2]School of Management, University College London,
Gower Street, London, WC1E 6BT, UK

[2]Institute for Systems and Computer Engineering, Technology and Science,
University of Porto (UP), Porto, Portugal
[2]Laboratory of Technology and Innovation (LATIN),
Federal Institute of South of Minas Gerais (IFSULDEMINAS), Poços de Caldas, Brazil

*E-mail: douglas.braz@ifsuldeminas.edu.br

This document contains additional information related to the original paper "*Forecasting Financial Market Structure from Network Features using Machine Learning*".

S.1 Pseudo-Algorithm of the Link Prediction Approach

This section presents the pseudo-algorithm used in the link prediction approach proposed in the original paper.

Algorithm 1 Link Prediction Approach

```
Require: G_T(V, E)
  k ← 30
  t ← k
  while t ≤ T do

    ## Creating the train set ##
    for i ← 1, k do
      node_features ← extractNodeFeatures(G_{t−i}(V, E))
      link_features ← extractLinkFeatures(G_{t−i}(V, E))
      train_set ← combine(train_set, concatenate(node_features, link_features))
    end for

    ## Creating the test set ##
    node_features ← extractNodeFeatures(G_t(V, E))
    link_features ← extractLinkFeatures(G_t(V, E))
    test_set ← concatenate(node_features, link_features)

    ## Training and testing the machine learning model ##
    model ← train(train_set)
    prediction ← predict(model, test_set)
    evaluate(prediction)

    t ← t + δT
  end while
```

S.2 Hyper Parameters of the Machine Learning Model

This section presents the set of hyper parameters used in XGboost algorithm to run the experiments. The configuration attributes are:

- booster = “gbtree”;

- objective = “binary:logistic”;
- eta = 0.05;
- max_depth = 4;
- min_child_weight = 120;
- max_delta_step = 0;
- subsample = 0.75;
- colsample_bytree = 0.99;

S.3 Network Persistence Results - Cross Similarity Matrices

This section presents the set of experimental results describing the financial network persistence over the period between 12 May 2006 and 18 December 2019. This analysis allows us to measure how the financial networks change their structure over time. We estimate the network persistence by calculating pair-wise similarity between networks $G(t)$ and $G(t')$ using the Jaccard Distance, defined as follows:

$$sim(G(t), G(t')) = \frac{|G(t) \cap G(t')|}{|G(t) \cup G(t)|}, \tag{1}$$

where t and t' range from 12 May 2006 to 18 December 2019.

Results for $L = 126$ trading days to create each graph are presented in Figures S1, S2 and S3. Results for $L = 504$ trading days are presented in Figures S4, S5 and S6.

S.4 Complementary Results and Analysis

This section provides experimental results using the size of rolling window $L = 126$ and $L = 504$ trading days to construct the financial networks. Considering $L = 126$, Figures S7, S8

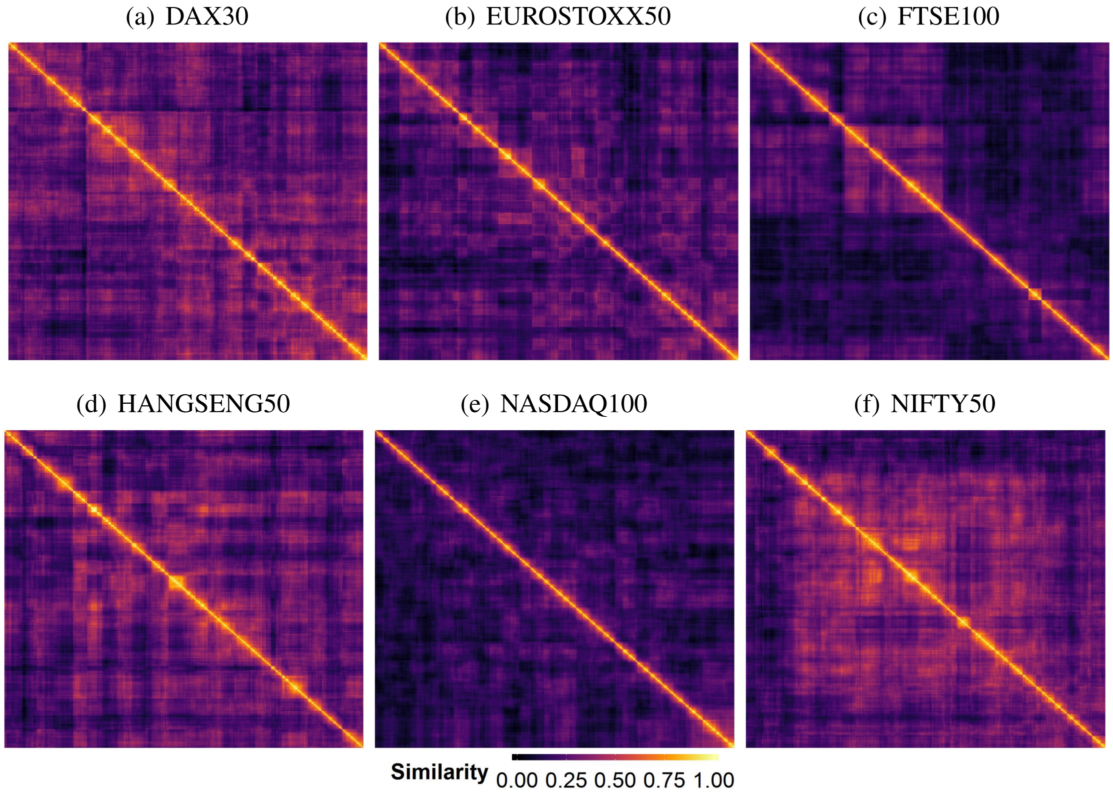

Fig. S1: **DAG - Cross-similarity matrix for each market index using $L = 126$.** We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom-right corner of each individual figure.

and S9 show the AUC measure of the proposed machine learning method compared against baseline algorithms for DAG, DTN and DMST network filtering methods. Considering $L = 504$, Figures S11, S12 and S13 present results for DAG, DTN and DMST network filtering methods, respectively. For each time step ahead h, we calculated the average AUC of each method and its respective standard error over the test period, ranging from 5 May 2007 to 18 December 2019.

Figures S10 and S14 present the AUC performance and the AUC* improvement of the proposed method using $L = 126$ and $L = 504$ for h trading weeks ahead ($1 \leq h \leq 20$). Results

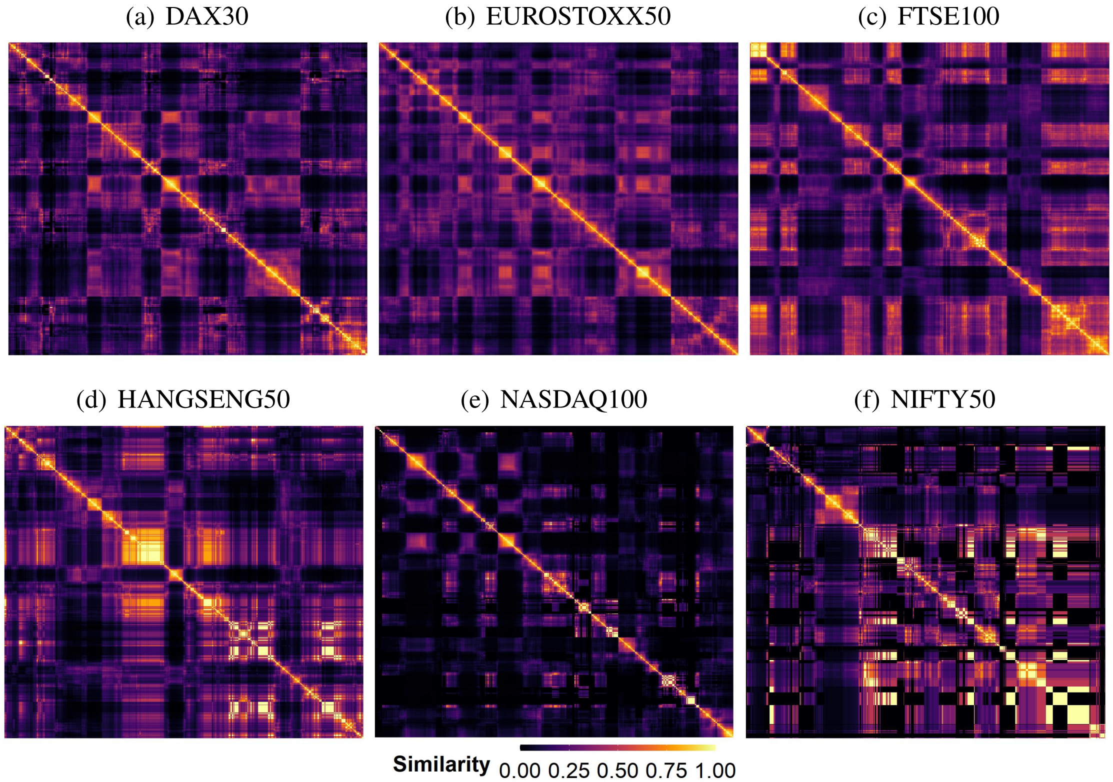

Fig. S2: **DTN - Cross-similarity matrix for each market index using $L = 126$.** We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom-right corner of each individual figure.

are provided for DAG, DTN and DMST network filtering methods. The AUC* improvement is calculated over the time invariant (TI) benchmark method.

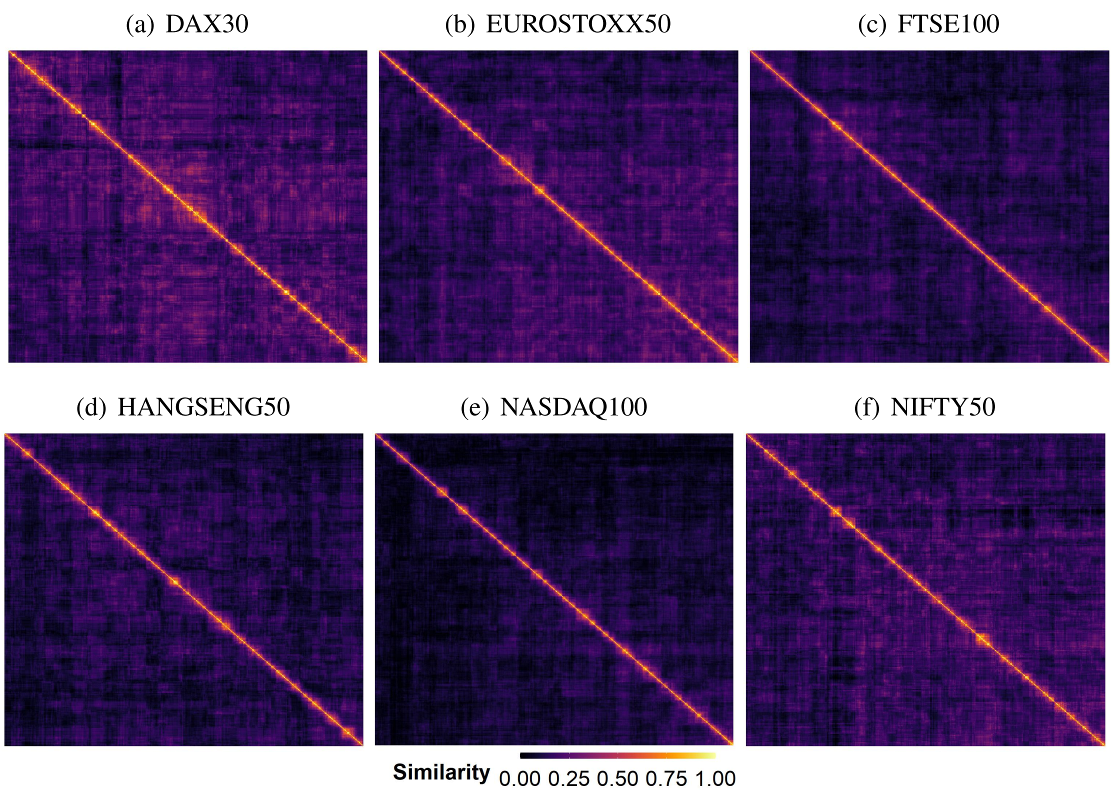

Fig. S3: **DMST - Cross-similarity matrix for each market index using $L = 126$.** We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom-right corner of each individual figure.

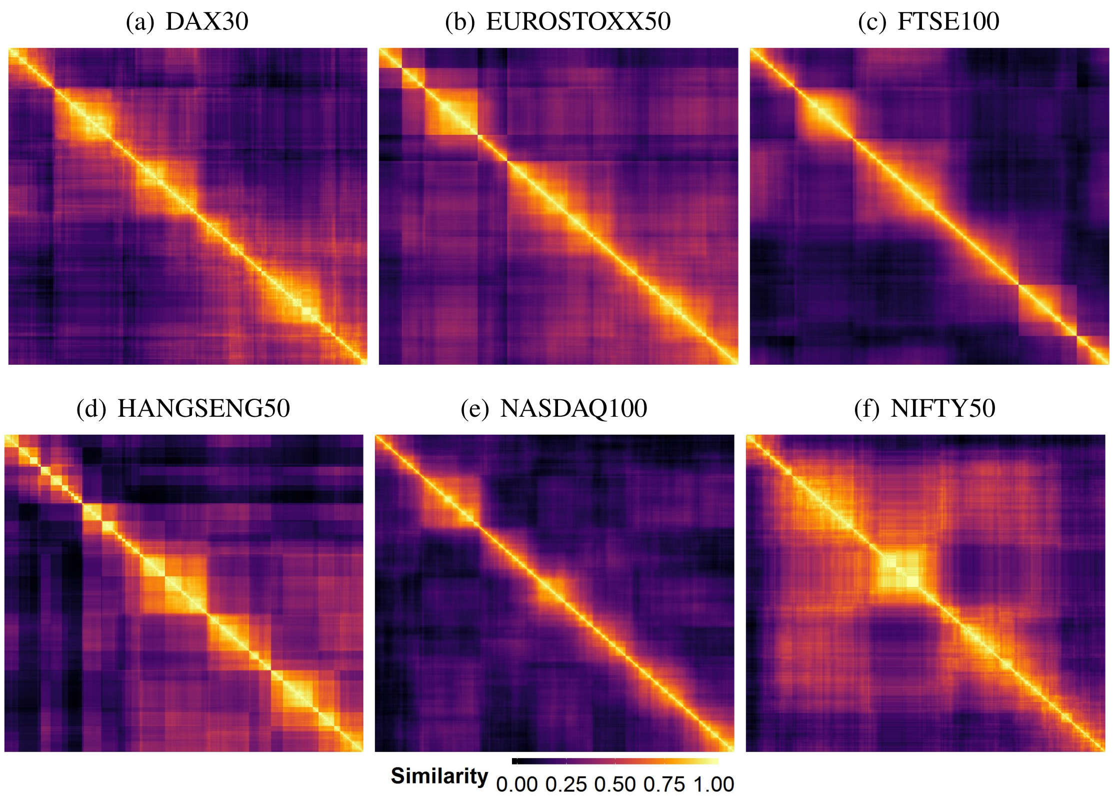

Fig. S4: **DAG - Cross-similarity matrix for each market index using** $L = 504$**.** We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom-right corner of each individual figure.

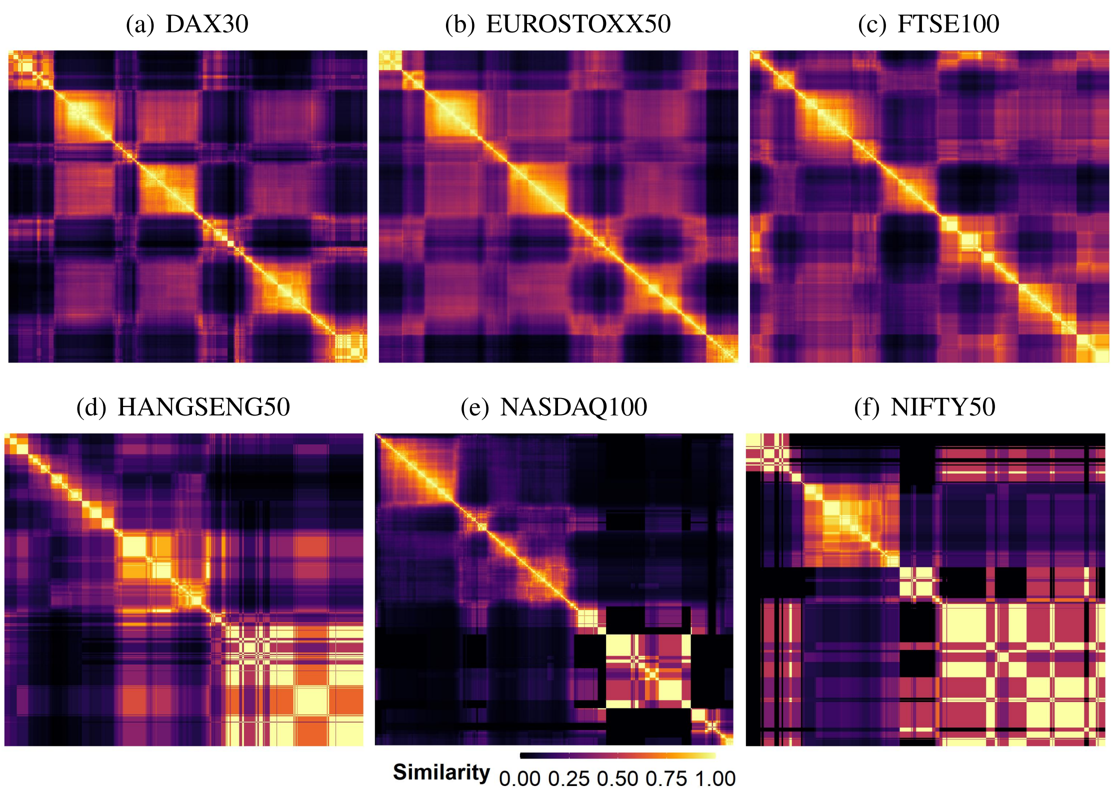

Fig. S5: **DTN - Cross-similarity matrix for each market index using** $L = 504$**.** We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom-right corner of each individual figure.

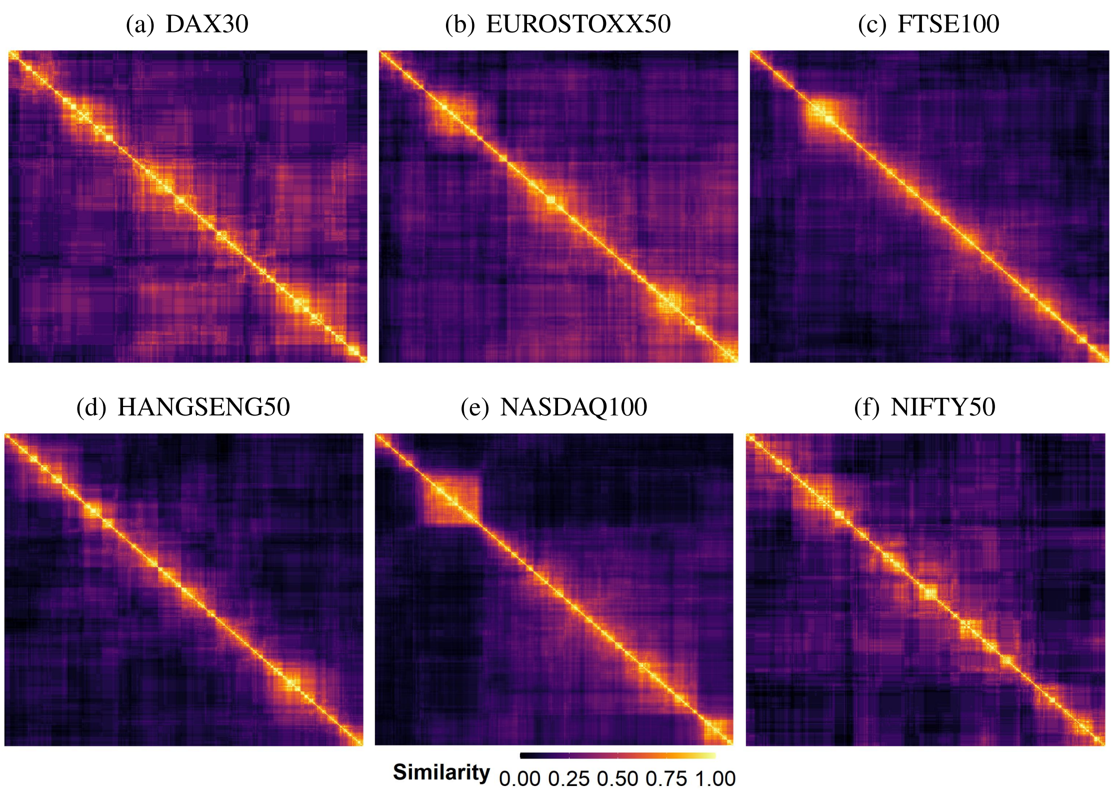

Fig. S6: **DMST - Cross-similarity matrix for each market index using $L = 504$.** We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom-right corner of each individual figure.

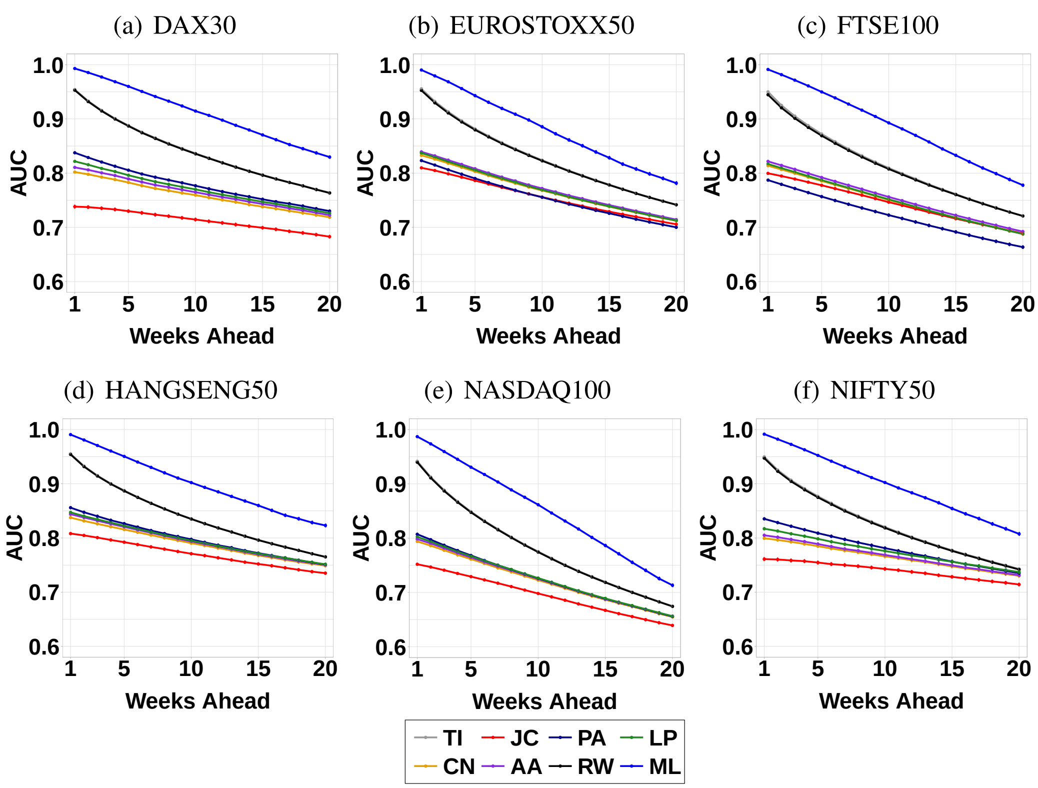

Fig. S7: **DAG - Predictive performance comparison of all methods.** Figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices.

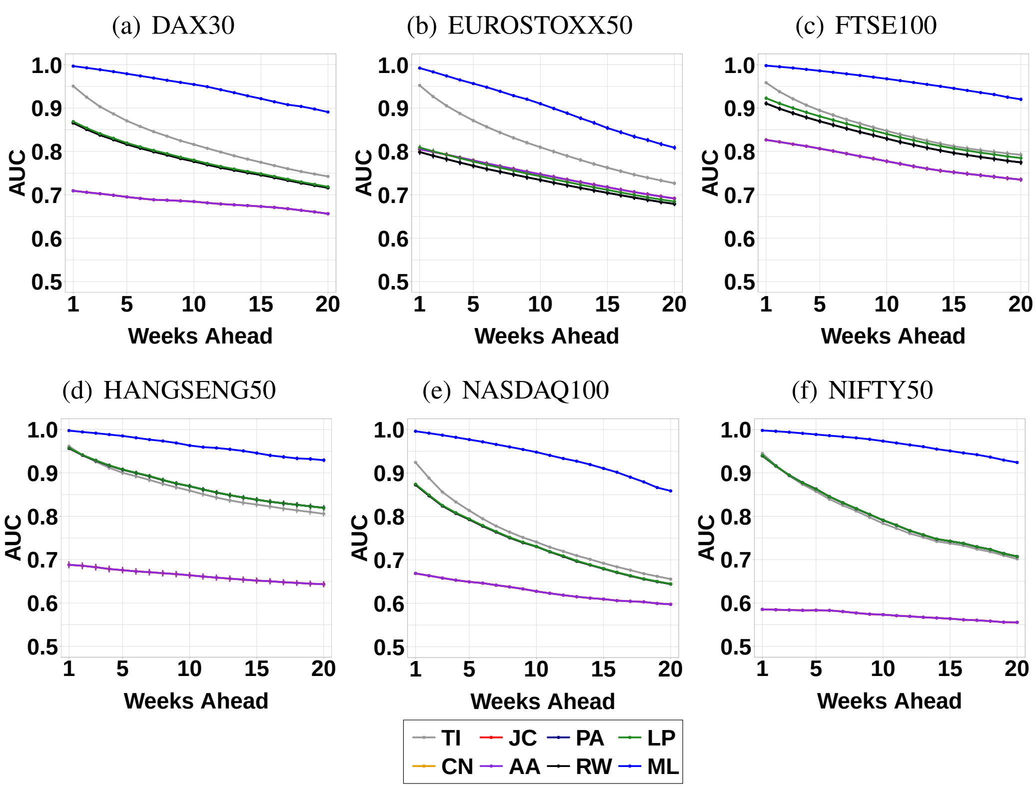

Fig. S8: **DTN - Predictive performance comparison of all methods.** Figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices.

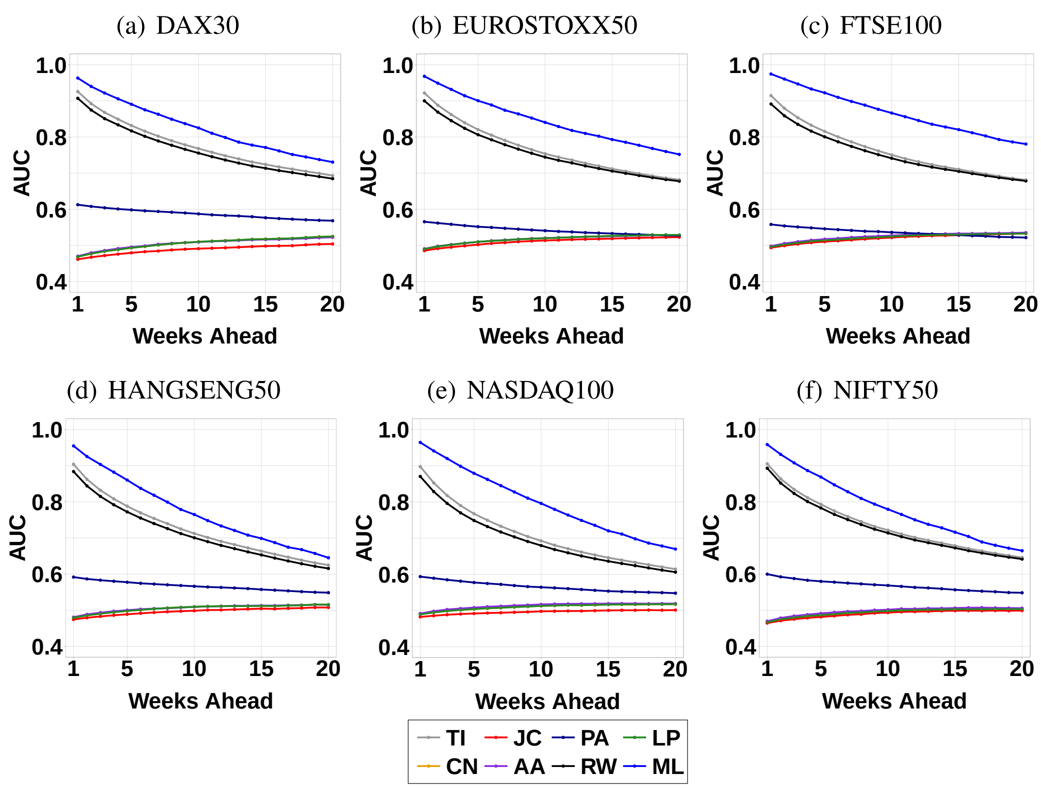

Fig. S9: **DMST - Predictive performance comparison of all methods.** Figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices.

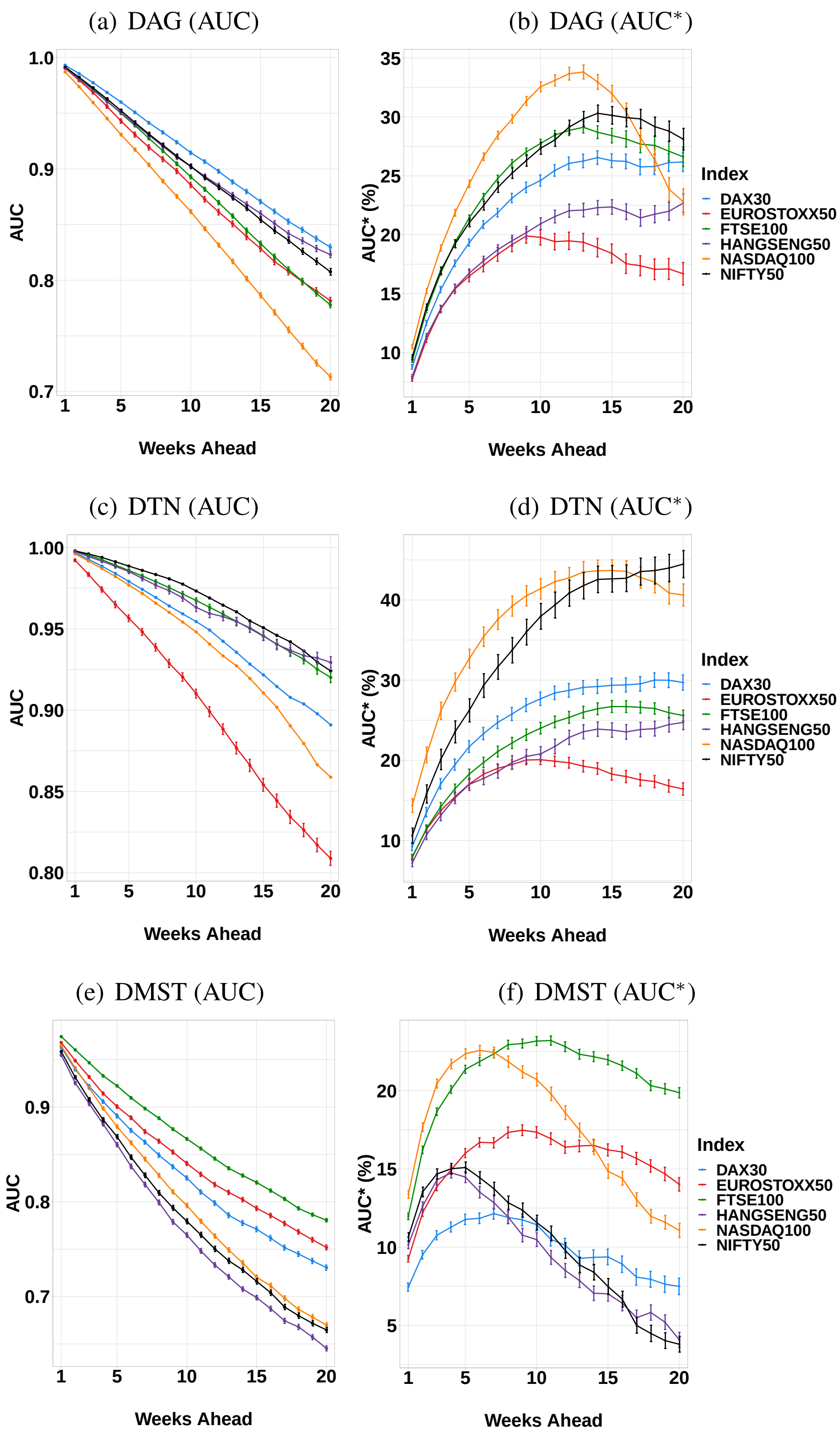

Fig. S10: **Machine Learning AUC and AUC* for DAG, DTN and DMST network filter methods.** Panels (a), (c) and (e) present the machine learning AUC measure and its standard error for h trading weeks ahead ($1 \leq h \leq 20$). Panels (b), (d) and (f) present the AUC improvement over the benchmark time-invariant method and its standard error. Results for $L = 126$.

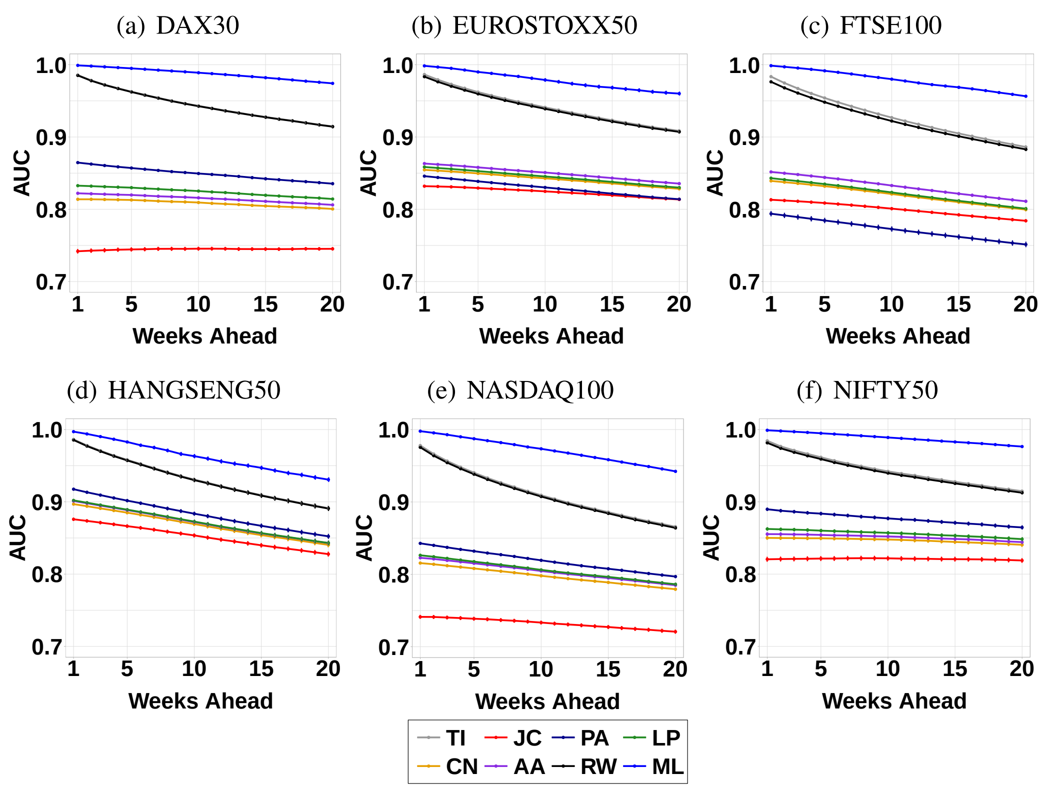

Fig. S11: **DAG - Predictive performance comparison of all methods.** Figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices.

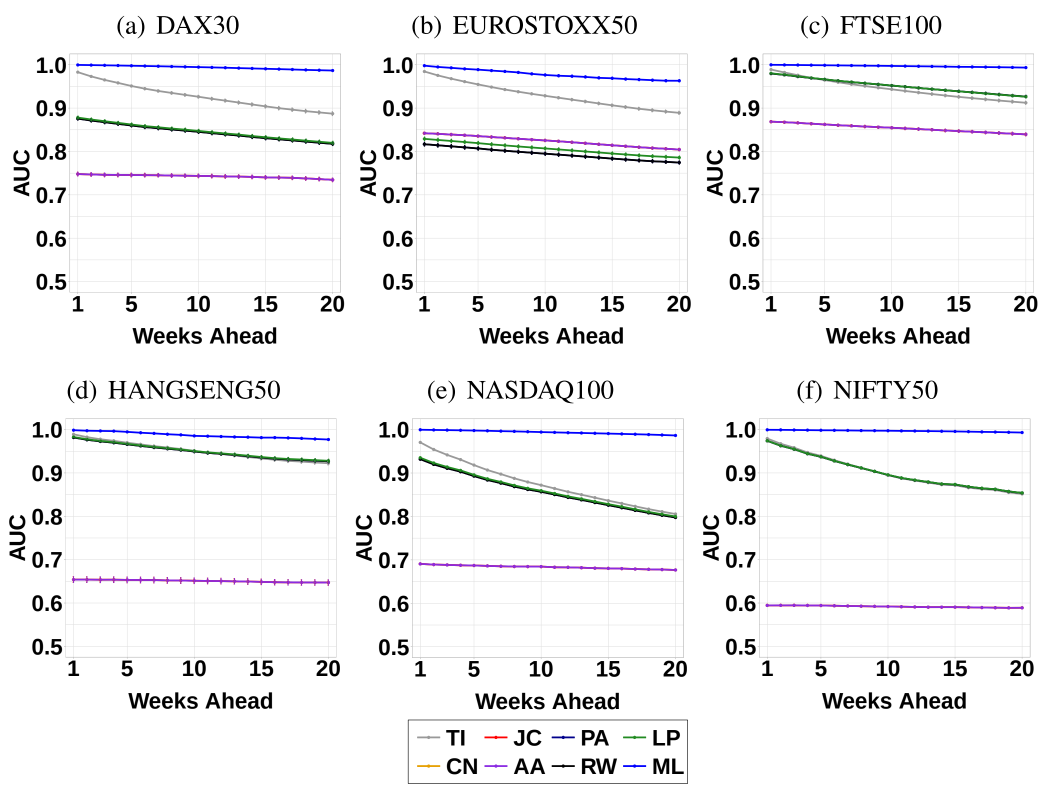

Fig. S12: **DTN - Predictive performance comparison of all methods.** Figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices.

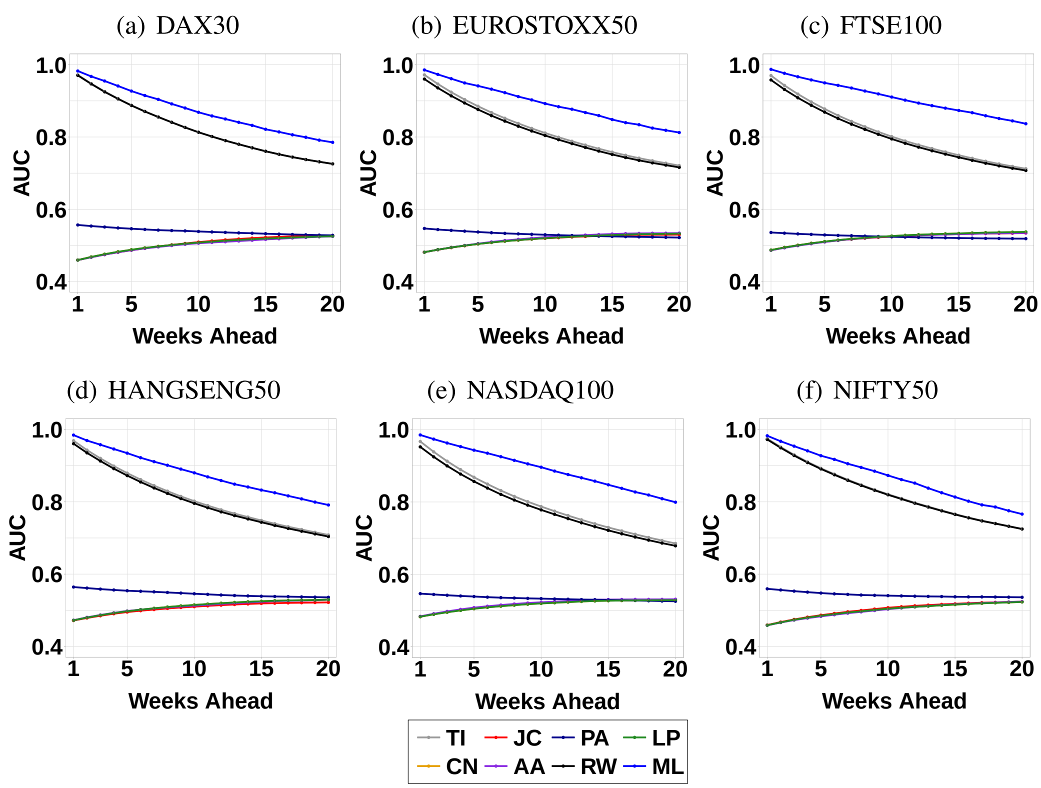

Fig. S13: **DMST - Predictive performance comparison of all methods.** Figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices.

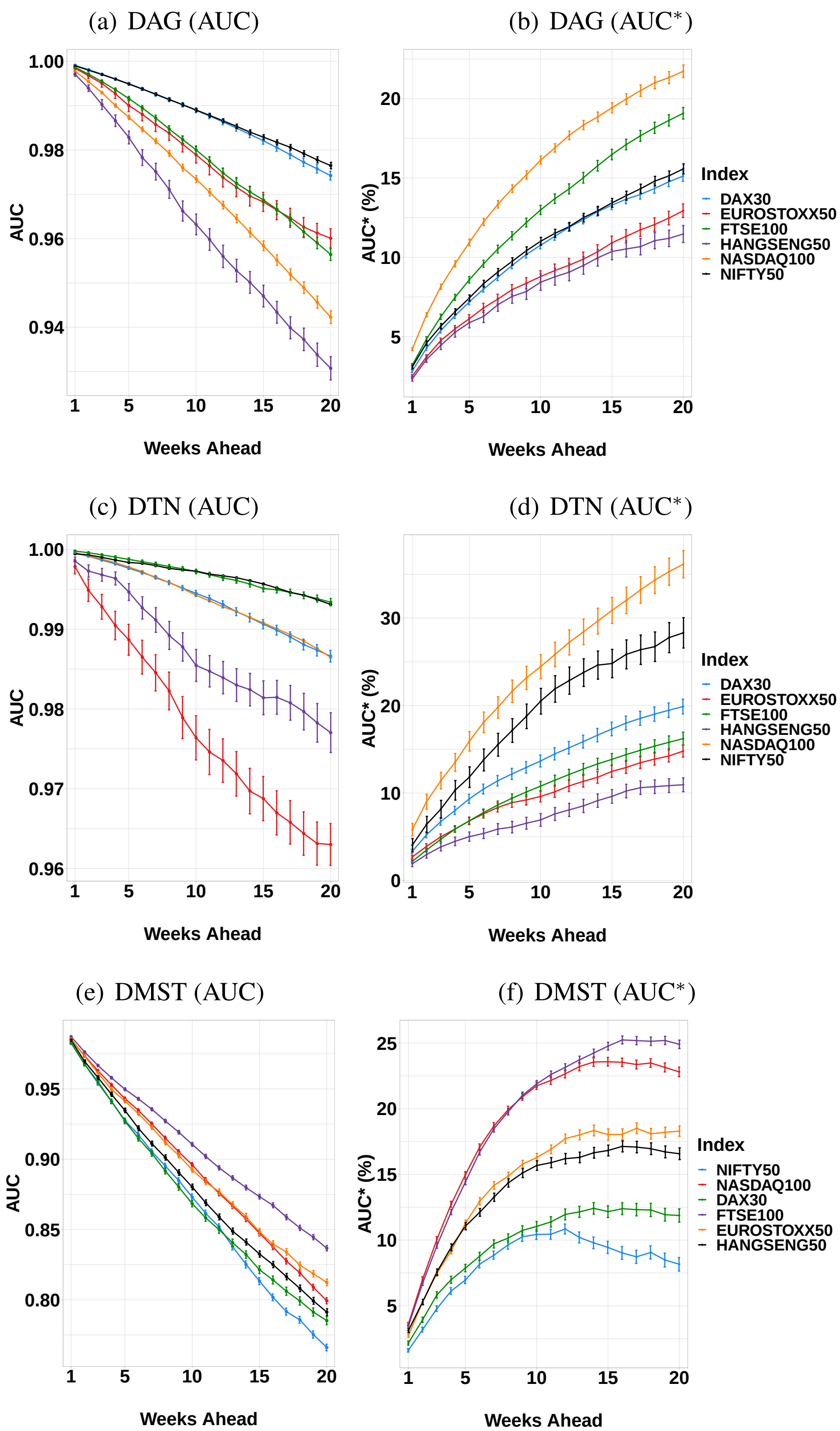

Fig. S14: **Machine Learning AUC and AUC* for DAG, DTN and DMST network filter methods.** Panels (a), (c) and (e) present the machine learning AUC measure and its standard error for h trading weeks ahead ($1 \leq h \leq 20$). Panels (b), (d) and (f) present the AUC improvement over the benchmark time-invariant method and its standard error. Results for $L = 504$.