"It Looks All the Same to Me": Cross-index Training for Long-term Financial Series Prediction

Stanislav Selitskiy[0000−0003−1758−0171]

University of Bedfordshire, Park Square, Luton, LU1 3JU, UK
stanislav.selitskiy@study.beds.ac.uk

Abstract. We investigate a number of Artificial Neural Network architectures (well-known and more "exotic") in application to the long-term financial time-series forecasts of indexes on different global markets. The particular area of interest of this research is to examine the correlation of these indexes' behaviour in terms of Machine Learning algorithms cross-training. Would training an algorithm on an index from one global market produce similar or even better accuracy when such a model is applied for predicting another index from a different market? The demonstrated predominately positive answer to this question is another argument in favour of the long-debated Efficient Market Hypothesis of Eugene Fama.

Keywords: Efficient Market Hypothesis · neural networks · cross-training.

1 Introduction

The Efficient Market Hypothesis (EMH) in the well-known form was popularized by Eugene Fama [4] in the late 60's - the early '70s, though the earliest documented high-level formulation was known since 16'th century [28]. In various forms of strictness, it postulates that all potentially available information is immediately incorporated into market prices. This quite radical and superficial (and, in a way, superstitious and superfluous) proposition was fiercely debated over time. The early discussion of the EMH hypothesis was primarily conducted by proponents of the hypothesis on the material of the developed markets, which led to over-confident conclusions. In the '80s, the application of the theory was tested on emergent markets, which brought mixed results. In the '90s, the challenge of the EMH became widespread, though, even in the 21st century, the discussion still continues [35].

The economic [21], environment [20], logical paradoxes [10], and psychological [29] aspects of the debate are out of the scope of this paper. Machine Learning (ML) and statistical market prediction methods based on the previous historical data are the "bread and butter" of the practising traders and one of the arguments favouring the EMH [33]. However, even though these algorithms have some limited predicted power in plain form, they do not strictly show the use of the information in the predictions but possibly stochastic correlations. Research

on tracing how publicly unknown information finds its way to the public and into market prices, the rate and cost of its dissipation is still to be conducted.

In this research, we still use the indirect method. We investigate the hypothesis that if today's global markets are affected by information, then their behaviour is correlated long-term. It is a trivial proposition, which has been debated for decades [3], but we give it a practical spin in our applied research, going beyond usual statistics correlation analysis [18]. If we train an ML model on one index (potentially on one market) and then use it for index prediction of another index on another global market, and those predictions have similar or better accuracy (due to less over-fitting), then this correlation would be an argument in favour to EMH.

In the study, we experiment with various Artificial Neural Network (ANN) architectures, the well-known easily assembled from the layers available in many ML libraries, more "exotic" that require additional manual coding, and developed from scratch by the authors, to ensure that cross-training effects are common and not bound to one particular ANN architecture. We show how successfully those ANN models work in long-term forecasting for 30 days when trained on the same index data and then when they are cross-trained. We use historical data from 2005 to the beginning of 2022 for the NASDAQ, Dow Jones, NIKKEI and DAX indexes to cover intra-market and cross-market effects.

The contribution is organized in the following way: Section 2 presents closely related to the study's existing work, Section 3 high-level describes ML algorithms chosen for the study, data sets and their partition, which were used in computational experiments, as well as accuracy metrics for algorithms' evaluation. Section 4 lists hardware parameters and model configurable parameters, as well as details of the less-popular ANN architectures implemented from scratch. Section 5 shows the results of the experiments in diagram and table form, discusses these results and study limitations, draws conclusions and outlines future research directions.

2 Related work

ML algorithms of different types were used for various markets, stock types, observation periods, and prediction intervals.

Two classes of the ML algorithms, Decision Tree (DT) based and ANN-based, were used on the 10 years interval for the Tehran Stock Exchange data (financial, petroleum, mineral, and metal indexes) in [14]. Various prediction intervals ranging from 1 to 30 days in the future were used, measuring accuracy with four metrics. Unfortunately, the best results were reported over the whole time span and prediction intervals, which makes the research of limited interest. However, systematically, Long Short-term Memory (LSTM) ANN was reported as superior to other ML models.

In [30], not just composite indexes, but instead, stocks of the 25 individual companies on the Bucharest Stock Exchange were experimented with on the span of over 21 years. Two ANN models, LSTM and 1-dimensional temporal

Convolutional Neural Network (CNN), were used not just for accuracy calculation but also for the trading gain simulation in comparison with traditional simple trading tactics. Predictions were run on training intervals from 30 to 120 days, but with only 1 day prediction in the future at a time. Both architectures demonstrated superior performance compared to the simple tactics specialising in the window frequency for CNN or gain amounts for LSTM.

Traditionally ML models are trained on the same indexes but on the previous chronological data. Of our particular interests are approaches of the ML models training on the different time series. "Cross-training" is not a well-established term; therefore, such methods come under various rubrics, such as ensembles, pre-training, and transfer learning. They include noise-augmented training data [36], contrastive self-supervised learning targeting on the extraction of the augmented components [34], averaged LSTM training on multiple index data [32], or averaging of the ensemble of the models trained on the multiple indexes [7].

Although some similarities could be seen between the above-mentioned and presented research, it is worth pointing out that the aim of the existing research is the enriching training data of the pre-trained transfer-learning models with the superset of the patterns found either in other indexes (or domain) or synthetic time series. In our study, ANN models are trained on the index of one single market and then tested on the other three markets, and such experiments are applied to all four markets in a circular fashion. No averaging or other use of data from multiple markets for training is done because the study aims to experimentally confirm the alleged by EMH synchronicity of the markets, strong enough to have no need for averaging or multiple data sets pre-training.

3 Methods

It could be easily seen in Figure 1, Table 1, the stock indexes used in this study and described in more detail in Section 3.2, are correlated.

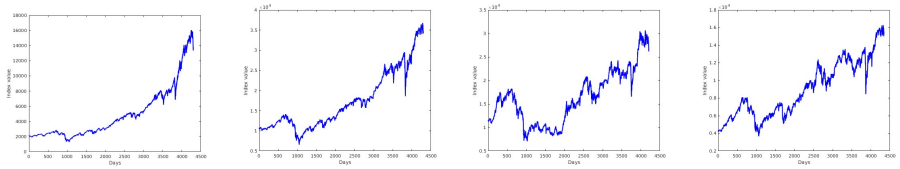

Fig. 1. Stock indexes NASDAQ, DOW, NIKKEI, DAX on intervals of the experiments.

However, how much of this correlation can be used for practical purposes of cross-training and cross-forecasting? Furthermore, how much of theoretical conclusions in the context of EMH can we extract? Remaining inside statistical methods, such cross-training is not intuitive, especially from the Frequentist's point of view, though the Bayesian can look at statistics obtained from another

Table 1. Pearson correlation coefficient, applied pairwise to the stock indexes under experimentation

Index 1	Index 2	Correlation
NASDAQ	NIKKEI	0.8879
NASDAQ	DOW	0.9750
NASDAQ	DAX	0.8975
DOW	DAX	0.9456
DOW	NIKKEI	0.9053
NIKKEI	DAX	0.8740

index as a starting belief to polish them on the target stock statistical forecasts using statistical [9, 22, 26] or ML methods [25]. Also, choosing particular statistics requires prior assumptions about patterns that can embed information the stocks are supposed to absorb by EMH.

ML methods, and especially ANNs', strength is freedom of necessity for such *a priory* assumptions. ANN can learn unexpected patterns, and if there are such patterns in the stock series which can be cross-learned and successfully used for cross-forecasting, that is an argument for EMH, considering the ubiquity of the information in global markets, economy, and the world itself. To leverage the best models mentioned in Section 2 and enhance them, we concentrated on ANN architectures, significantly increasing the variety of the being studied models and applying them for the extreme for the cited works prediction interval of 30 days.

Accuracy metrics (described below in Subsection 3.1) and their distribution over session partitions (also described below in Subsection 3.2), are collected for all being investigated ANN models. As ablation testing, prediction accuracies are calculated in the absence of cross-training (the same index is used for training and testing, but on different time intervals). Acquired accuracy distributions for cross-training are subjected to the non-parametric hypothesis testing Wilcoxon signed rank algorithm to verify that non-cross-training and cross-training accuracy distribution either have no shift or "right" (greater) shift for non-cross-training distributions, i.e. cross-training accuracy is at least no worse than the non-cross-training.

3.1 Accuracy metrics

As an accuracy metrics, we use the Mean Absolute Percentage Error (MAPE), defined as follows:

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} |\frac{A_t - F_t}{A_t}| \tag{1}$$

And Root Mean Square Error (RMSE):

$$RMSE = (\frac{1}{n} \sum_{t=1}^{n} (A_t - F_t)^2)^{\frac{1}{2}} \tag{2}$$

Where A_t and F_t are the actual and predicted indexes at a given day t, respectively, and n is the number of test observations.

3.2 Data sets

The data for NASDAQ, Dow Jones, NIKKEI, and DAX indexes for the beginning of 2005 year up to the end of January 2022 were used in computational experiments. The data were downloaded from `https://tradingeconomics.com`. Detailed statistical analysis of the data set is out of the scope of the paper; however, general intuition about this strongly non-stationary time series can be obtained from general Figure 1, and simple Pearson correlation coefficients Table 1. The data represent a broad spectrum of behavioural tendencies: oscillations of the stagnant markets, explosive growth and fall during bubble bursts and busts in American, European, and Asian markets. The difference in stock exchanges' working days schedules made the data slightly (a few days) asynchronous. The reason for the particular indexes selection is an attempt to limit the combinatorial complexity of the experiments (only four indexes), still preserving geographical representability (at least for the most economically important markets: DOW, NIKKEI, DAX), and general vs. industry-specific markets (DOW and NASDAQ).

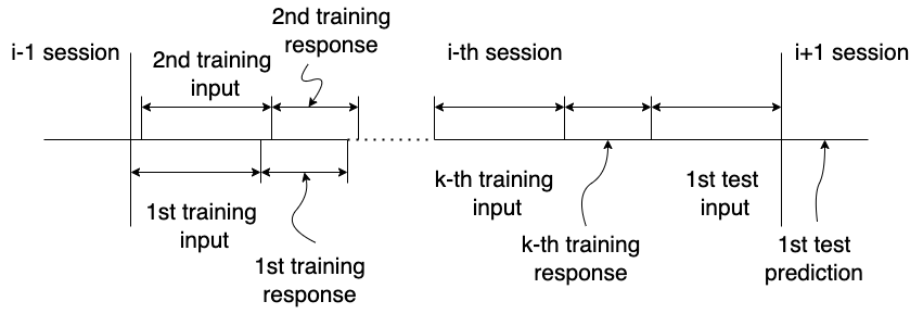

Fig. 2. Session partition schema.

To accommodate long-term prediction of 30 days forward, based on the past 30 days indexes performance, the data were divided into 35 subsets, each consisting of 30 observations used to train the sequence-to-sequence models. Each "observation" was comprised of 30 days values and the output "label" values were the $31^{th} - 60^{th}$ days. Each following observation starts with a 1 day shift. The number of training sessions is first 34, and the whole session is 120 days. It was defined so that none of the training data (including the label days) would touch the following test session data. The number of test sessions is last 34. The last 30 days of the preceding training session (not used in the training process) were used to predict the first 30 days of the following test session, with step 1 until

the end of the test session. Models' parameters were reset for each session, and training was done anew (see 2). For sequence-to-value models, such as LSTM, the whole 119 + 1 last label day was used in the training session.

4 Experiments

The experiments were run on the Linux (Ubuntu 20.04.3 LTS) operating system with two dual Tesla K80 GPUs (with 2×12GB GDDR5 memory each) and one QuadroPro K6000 (with 12GB GDDR5 memory, as well), X299 chipset motherboard, 256 GB DDR4 RAM, and i9-10900X CPU. Experiments were run using MATLAB 2022a with Deep Learning Toolbox. For inferential statistics, built-in R 4.2.1 implementations were used with default parameters unless mentioned otherwise.

Table 2. Accuracy of various ML models for NASDAQ index

Model	MAPE	RMSE
AR	0.0521 ± 0.0428	336.79 ± 471.61
ANN reg.	0.0504 ± 0.0401	320.75 ± 405.89
Logistic	0.0541 ± 0.0417	313.21 ± 315.77
ReLU	0.0560 ± 0.0421	361.88 ± 432.68
LSTM vec.	0.0506 ± 0.0273	283.15 ± 229.51
LSTM	0.0615 ± 0.0590	316.60 ± 274.46
SCNN	0.0643 ± 0.0590	360.77 ± 358.64
RBF	0.0588 ± 0.0418	343.20 ± 336.99
KGate	0.0623 ± 0.0555	363.56 ± 351.92
GMDH	0.0549 ± 0.0505	316.23 ± 321.70

The following models were experimented with: analytically-solved multi-variate linear Auto-regression, ANN Regression (no activation functions), ANN with ReLU (Rectified Linear Unit), Logistic, Hyperbolic tangent activations, sequence-to-sequence and sequence-to-value LSTM and GRU (Gated Recurrent Unit), simple sequential and spectral cascade CNN, RBF (Radial Basis Function), KGate (Kolmogorov's Gate), and finally GMDH (Group Method of Data Handling) ANNs. Because ANN regression, ANN with ReLU, KGate activations and CNNs are tolerable to non-normalized input data and frequently produce better accuracy, for these models, experiments are done with non-normalized input. For other models sensitive to normalization, a min-max normalization is applied in a strict mode, calculated only for a given observation data without looking ahead of time or in the past. A mean squared error was used as a loss function for all ANNs.

4.1 Ready-to-use ANN layers

Well-known ANN layers and architectures, readily available in MATLAB or its Toolboxes, are not described here in detail - only non-default configuration pa-

rameters are presented below. Non-standard implementations and the whole source code used in experiments can be found on GitHub (`https://github.com/Selitskiy/LTTS`) and are implemented as follows.

If we look at linear regression, or the linear part of a layer transformation of an ANN, as a transformation from a higher dimensional space into the lower dimensional space f, $n < m$:

$$f : \mathcal{X} \subset \mathbb{R}^m \mapsto \mathcal{Y} \subset \mathbb{R}^n \tag{3}$$

Linear regression can be represented as a matrix multiplication:

$$\mathbf{y} = f(\mathbf{x}) = \mathsf{W}\mathbf{x},\ \forall \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^m,\ \forall \mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^n \tag{4}$$

where $\mathsf{W} \in \mathcal{W} \subset \mathbb{R}^{n \times m}$ is the adjustable coefficient matrix that, using minimization of the sum of squared errors, can be found as [6]:

$$\mathbf{W} = (\hat{\mathbf{X}}^T \hat{\mathbf{X}})^{-1} \hat{\mathbf{X}}^T \hat{\mathbf{Y}} \tag{5}$$

where $\hat{\mathbf{X}}$, $\hat{\mathbf{Y}}$ are matrices of the observations of the input and output observations, respectively [23].

All other ANN architectures, to compare them on a similar level of complexity (number of learnable parameters), were designed to have two hidden layers with a number of neurons m and $2m + 1$ on the first and second hidden layer, respectively, where m is an input dimensionality, Formula 3. The reason to limit ANN models to two layers was if one looks at ANN as a Universal Approximation according to the Kolmogorov-Arnold superposition theorem [11], for general emulation of the $f : \mathcal{X} \subset \mathbb{R}^m \mapsto \mathcal{Y} \subset \mathbb{R}$ process, the 2-layer is a minimal ANN configuration needed (given that activation functions are complex enough):

$$f(\mathbf{x}) = f(x_1, \ldots, x_m) = \sum_{q=0}^{2m} \Phi_q \left(\sum_{p=1}^{m} \phi_{qp}(x_p) \right) \tag{6}$$

where Φ_q and ϕ_{qp} are continuous $\mathbb{R} \mapsto \mathbb{R}$ functions.

The practicality of such ANN, as a Universal Approximator, was disputed in [5], particularly because of the non-smoothness, hence non-practicality, of the inner ϕ_{qp} functions. However, these objections were rebutted in [13]. In [17] ϕ_{qp} activation functions are even called "pathological".

Therefore, in addition to the usual activation functions, much less frequently used architectures and activation functions were experimented with.

4.2 Custom-coded and originally-developed ANN layers

Radial Basis Functions (RBF) ANN were proposed at the end of the 80s [2]. It could be viewed as a "soft gate" which activates the transformation matrix coefficients in Gaussian proportion to the proximity of the test signal to the training signals, this transformation matrix's coefficients were trained at [16].

Table 3. Accuracy of various ML models for NIKKEI index

Model	MAPE	RMSE
AR	0.0808 ± 0.0723	1919.10 ± 2707.51
ANN reg.	0.0799 ± 0.0723	1877.54 ± 2630.55
Logistic	0.0890 ± 0.0798	1963.34 ± 2596.88
ReLU	0.0904 ± 0.0989	2256.28 ± 3905.08
LSTM vec.	0.0598 ± 0.0248	1165.07 ± 467.01
LSTM	0.0587 ± 0.0286	1120.97 ± 519.30
SCNN	0.0932 ± 0.0924	2122.16 ± 3200.82
RBF	0.0923 ± 0.0818	2105.33 ± 3015.20
KGate	0.1008 ± 0.1086	2386.27 ± 3854.98
GMDH	0.0932 ± 0.0926	2087.34 ± 2806.60

$$f(\mathbf{x}) = \mathbf{a}_k e^{-\mathbf{b}_k(\mathbf{x}-\mathbf{c}_k)^2} \tag{7}$$

where $k \in \{1 \dots n\}$.

An apparent drawback of the architecture is its "fluffiness" due to the non-reuse of the neurons for the "missed" test-time data input, making the RBF ANNs less dense or compact than Deep ReLU ANNs. Still, RBF is a viable architecture used in niche applications [12, 1].

Another ANN architecture [24] can be seen as a part of the Gated Linear Unit (GLU) family of activations. Using Directed Acyclic Graph (DAG) ANN, one can implement a cell (let us call it Kolmogorov's Gate or KGate for short) of perceptrons with logistic sigmoid activations that would work as allow or do not allow gates at saturation domain or multiplicative scaling of the main trunk of ANN, in the non-saturation domain of input values. Perceptrons with hyperbolic tangent activation would work as update/forget or the mean shift gates on the main ANN trunk, working together with the linear input transformation through the multiplication gate, Formula 8.

$$\begin{aligned} &\mathbf{z}_i = (\mathsf{W}_i\mathbf{x}_i + (\tau \circ \mathsf{W}_{ti}\mathbf{x}_0) \odot (\mathsf{W}_{ai}\mathbf{x}_0)) \odot \sigma \circ \mathsf{W}_{si}\mathbf{x}_0, \\ &\forall \mathbf{x}_0 \in \mathcal{X}_0 \subset \mathbb{R}^m, \forall \mathbf{x}_i \in \mathcal{X}_i \subset \mathbb{R}^{m_i} \end{aligned} \tag{8}$$

where $\mathbf{x}_0$ is an ANN input, $\mathbf{x}_i$ is an input of the i^{th} layer, $\mathsf{W}_i\mathbf{x}_i$ is the linear transformation of the main trunk, $\mathsf{W}_{ti}\mathbf{x}_0$, $\mathsf{W}_{ai}\mathbf{x}_0$, $\mathsf{W}_{si}\mathbf{x}_0$ are linear transformations inside the KGate cell, and τ, σ are hyperbolic tangent and logistic sigmoid activation functions, respectively.

Following Ivakhnenko [8], the multi-layer neural-network models could be grown by the Group Method of Data Handling (GMDH) using a neuron activation function defined by a short-term polynomial, in our case - the polynomial of the second degree of the pairwise connected neurons, which ensures linear gradient optimization hyperplane on each layer generation step [15].

$$f(x_i, x_j) = (w_{ki}x_i + w_{kj}x_j + w_{k0})^2 \tag{9}$$

The GMDH is capable of generating new layers capable of predicting new data most accurately. The GMDH generates new neurons to be fitted to the training data in each layer. A given number of the best-fitted neurons are selected for the next layer.

Spectral cascade CNN (SCNN) was organized as parallel Directed Acyclic Graph (DAG) cells, similar to Inception or ResNet type cells [31, 19], of 3×1, 5×1, 7×1, 11×1, 13×1 dimensions, packets of 16 of each [27].

ANN models were trained using the "adam" learning algorithm with 0.01 initial learning coefficient, mini-batch size 32, and 1000 epochs.

5 Results

As mentioned above, computational experiments were conducted on NASDAQ, Dow Jones, NIKKEI, and DAX indexes partitioned as described in Section 3.2 using linear Auto-regression, ANN Regression, ANN with ReLU, Logistic, Hyperbolic tangent activations, sequence-to-sequence and sequence-to-value LSTM and GRU, simple sequential and spectral cascade CNN, RBF, KGate, Transformer-like non-linearity, and GMDH ANNs.

Table 4. MAPE of cross-trained LSTM sequence-to-vector model (rows - trained on the index, columns - tested on the index).

Index	NASDAQ	DOW	NIKKEI	DAX
NASDAQ	0.0506 ± 0.0273	0.0409 ± 0.0269	0.0574 ± 0.0258	0.0523 ± 0.0239
DOW	0.0501 ± 0.0261	0.0417 ± 0.0270	0.0589 ± 0.0280	0.0546 ± 0.0268
NIKKEI	0.0521 ± 0.0284	0.0416 ± 0.0243	0.0598 ± 0.0248	0.0551 ± 0.0243
DAX	0.0522 ± 0.0291	0.0425 ± 0.0283	0.0615 ± 0.0291	0.0589 ± 0.0377

Logistic and Hyperbolic tangent activation ANNs, LSTM and GRU, KGate and Transformer, CNN and SCNN produced similar results; therefore, only one of them is shown. For ablation non-cross-training accuracy data, only NASDAQ and NIKKEI index results are shown as extremes. To save space, Dow Jones and DAX results are not shown here Table 2, Table 3), but can be downloaded from the same GitHub link as the source code.

Table 5. MAPE of cross-trained RBF model (rows - trained on the index, columns - tested on the index).

Index	NASDAQ	DOW	NIKKEI	DAX
NASDAQ	0.0588 ± 0.0418	0.0485 ± 0.0393	0.0756 ± 0.0397	0.0590 ± 0.0316
DOW	0.1328 ± 0.4250	0.1020 ± 0.3150	0.0709 ± 0.0344	0.4299 ± 2.1876
NIKKEI	0.0698 ± 0.0680	0.0544 ± 0.0619	0.0923 ± 0.0818	0.0592 ± 0.0354
DAX	0.0649 ± 0.0503	0.0542 ± 0.0521	0.0702 ± 0.0339	0.0794 ± 0.1053

Results of the cross-training experiments are shown for all indexes, but also for extremes of the models - LSTM sequence-to-vector being most accurate, and RBF activation being most vulnerable for volatility: Table 4, Table 5.

P-values of the paired Wilcoxon signed rank test between non-cross-trained and cross-trained accuracy distributions for various ANN architectures for NASDAQ, Dow Jones, NIKKEI indexes, Table 6, 7, 8. Again, to save space, DAX results, which are similar to NASDAQ, are not shown.

Table 6. Wilcoxon signed rank test p-values on accuracy distribution over 34 sessions for models trained on NASDAQ data and tested on other indexes. P-values for alternative hypotheses of distribution shift for the not-cross-trained relative to cross-trained to the right, left, and two-sided.

Model	Index	Greater	Less	Two-side
Reg	DOW	0.00003	0.99997	0.00007
Reg	NIKKEI	0.97359	0.02757	0.05513
Reg	DAX	0.89874	0.10434	0.20869
ANN	DOW	0.00003	0.99997	0.00006
ANN	NIKKEI	0.97683	0.02421	0.04843
ANN	DAX	0.90761	0.09528	0.19056
ReLU	DOW	0.00001	0.99999	0.00001
ReLU	NIKKEI	0.93946	0.06276	0.12552
ReLU	DAX	0.53362	0.47310	0.94619
Sig	DOW	0.00160	0.99850	0.00319
Sig	NIKKEI	0.85610	0.14799	0.29599
Sig	DAX	0.81975	0.18476	0.36953
LSTMv	DOW	0.00000	0.99999	0.00000
LSTMv	NIKKEI	0.86407	0.13988	0.27976
LSTMv	DAX	0.68786	0.31815	0.63630
LSTM	DOW	0.00004	0.99996	0.00009
LSTM	NIKKEI	0.99013	0.01038	0.02077
LSTM	DAX	0.49327	0.51346	0.98654
GMDH	DOW	0.00004	0.99996	0.00009
GMDH	NIKKEI	0.79701	0.20805	0.41609
GMDH	DAX	0.75591	0.24945	0.49890
KGate	DOW	0.00003	0.99997	0.00006
KGate	NIKKEI	0.94161	0.06054	0.12109
KGate	DAX	0.49327	0.51346	0.98654
RBF	DOW	0.00069	0.99936	0.00137
RBF	NIKKEI	0.97683	0.02421	0.04843
RBF	DAX	0.55369	0.45299	0.90598
SCNN	DOW	0.00000	0.99999	0.00001
SCNN	NIKKEI	0.76010	0.24545	0.49089
SCNN	DAX	0.61937	0.38708	0.77417

6 Discussion and Conclusions

Interesting observations for ablation experiments are based on the MAPE metric, giving a more universal measure of the overall accuracy for each prediction point across the multiple indexes. In contrast, RMSE, though more index-specific, penalizes outlier predictions more, even though few of them exist. ANN models with exponential and polynomial non-linearities are especially vulnerable to stock volatility especially observed in NIKKEI and DAX indexes behaviour, which is one of the faces of the Out-of-Distribution (OOD) problem. Among all ANN architectures, the most accurate and robust was LSTM sequence-to-vector architecture.

Table 7. Wilcoxon signed rank test p-values on accuracy distribution over 34 sessions for models trained on DOW data and tested on other indexes. P-values for alternative hypotheses of distribution shift for the not-cross-trained relative to cross-trained to the right, left, and two-sided.

Model	Index	Greater	Less	Two-side
Reg	NASDAQ	0.99988	0.00013	0.00027
Reg	NIKKEI	0.99993	0.00007	0.00015
Reg	DAX	0.99948	0.00056	0.00112
ANN	NASDAQ	0.99979	0.00023	0.00047
ANN	NIKKEI	0.99996	0.00005	0.00009
ANN	DAX	0.99952	0.00052	0.00104
ReLU	NASDAQ	0.99980	0.00022	0.00043
ReLU	NIKKEI	0.99995	0.00006	0.00012
ReLU	DAX	0.99977	0.00025	0.00050
Sig	NASDAQ	0.99940	0.00064	0.00128
Sig	NIKKEI	0.99854	0.00156	0.00311
Sig	DAX	0.99272	0.00766	0.01531
LSTMv	NASDAQ	0.99999	0.00001	0.00001
LSTMv	NIKKEI	0.99993	0.00008	0.00016
LSTMv	DAX	0.99999	0.00001	0.00002
LSTM	NASDAQ	0.99999	0.00001	0.00002
LSTM	NIKKEI	0.99952	0.00051	0.00103
LSTM	DAX	0.98755	0.01304	0.02609
GMDH	NASDAQ	0.99830	0.00181	0.00361
GMDH	NIKKEI	0.99956	0.00048	0.00095
GMDH	DAX	0.94564	0.05631	0.11262
KGate	NASDAQ	0.99992	0.00009	0.00018
KGate	NIKKEI	0.99995	0.00006	0.00012
KGate	DAX	0.99940	0.00064	0.00128
RBF	NASDAQ	0.99999	0.00001	0.00003
RBF	NIKKEI	0.99742	0.00274	0.00548
RBF	DAX	0.99272	0.00766	0.01531
SCNN	NASDAQ	0.99993	0.00008	0.00015
SCNN	NIKKEI	0.99969	0.00033	0.00065
SCNN	DAX	0.99019	0.01029	0.02058

Table 8. Wilcoxon signed rank test p-values on accuracy distribution over 34 sessions for models trained on NIKKEI data and tested on other indexes. P-values for alternative hypotheses of distribution shift for the not-cross-trained relative to cross-trained to the right, left, and two-sided.

Model	Index	Greater	Less	Two-side
Reg	NASDAQ	0.00137	0.99872	0.00273
Reg	DOW	0.00009	0.99991	0.00019
Reg	DAX	0.01536	0.98536	0.03071
ANN	NASDAQ	0.00128	0.99881	0.00255
ANN	DOW	0.00007	0.99993	0.00015
ANN	DAX	0.01851	0.98233	0.03701
ReLU	NASDAQ	0.00128	0.99881	0.00255
ReLU	DOW	0.00009	0.99992	0.00017
ReLU	DAX	0.00685	0.99351	0.01369
Sig	NASDAQ	0.00017	0.99984	0.00034
Sig	DOW	0.00001	0.99999	0.00001
Sig	DAX	0.00097	0.99909	0.00195
LSTMv	NASDAQ	0.04666	0.95510	0.09332
LSTMv	DOW	0.00009	0.99991	0.00019
LSTMv	DAX	0.15642	0.84783	0.31283
LSTM	NASDAQ	0.36879	0.63785	0.73757
LSTM	DOW	0.01148	0.98908	0.02295
LSTM	DAX	0.25674	0.74894	0.51348
GMDH	NASDAQ	0.00214	0.99799	0.00428
GMDH	DOW	0.00038	0.99965	0.00076
GMDH	DAX	0.01396	0.98670	0.02791
KGate	NASDAQ	0.00112	0.99896	0.00223
KGate	DOW	0.00001	0.99999	0.00002
KGate	DAX	0.00026	0.99976	0.00051
RBF	NASDAQ	0.00177	0.99834	0.00354
RBF	DOW	0.00003	0.99997	0.00007
RBF	DAX	0.00035	0.99967	0.00070
SCNN	NASDAQ	0.01536	0.98536	0.03071
SCNN	DOW	0.00000	1.00000	0.00000
SCNN	DAX	0.00097	0.99909	0.00195

It also should be noted that this study is not about transfer learning - the approach popular in image recognition, when an ANN model trained on as much as possible large data is slightly modified and rapidly retrained for recognition of images from other domains, because it has already learned common image micro-patterns. That approach could have been used for stock indexes when time series from other domains, for example, weather prediction, are retrained specifically for financial series. However, that is not the aim of the study - we principally do not allow any retraining, hoping to find synchronous information-shared invariants in the untouched cross-trained models.

Results of the Wilcoxon tests (Tables 6-8) look mixed, with cross-trained accuracy being consistently better across all ANN models for NIKKEI index,

consistently worse across all ANN models for Dow Jones, and mixed for NASDAQ and DAX. However, the results are shown for 0 expected shift. If the shift is bounded by 0.02 (which is less than the standard deviation of even the best and narrowest accuracy distributions), then all models and stocks demonstrate at least no worse accuracy of the cross-trained models compared to the ablation non-cross-trained ones.

As a conclusion for practical purposes, if the bounded to the circa standard deviation, degradation of the cross-trained modes is accepted, in such a Lipschitz sense, the study supports a weak form of EMH, indirectly supporting the idea of existing of the patterns common for global markets and different stock types, which is allegedly maps to information in EMH context, of course in the study limitations frames. Another observation arguing in favour of the information nature of EMH is the tendency of larger indexes to "explain" smaller indexes, i.e. NIKKEI and NASDAQ vs DOW and DAX.

From the general ML point of view, considering local stock indexes as observations of the global process (world market), this study offers experimental validation of the plausible, but still speculations, that ML model trained on some local observations may effectively predict other synchronous observations.

6.1 Limitations

The study has obvious limitations in the geographic coverage and representation of only developed economies' stock exchanges. An apparent technical limitation of the study is using ANN models only, skipping other ML and statistical approaches. Another technical one, imposed by the hardware limitations, is the relative simplicity of the ANNs being used, barely fitting in the theoretical minimum of the universal approximation. The more subject-related limitation is that accuracy was calculated uniformly along the whole time span, without concentrating on the times of perturbations when a lot of new information allegedly entered the market.

6.2 Future Work

As an area for future research in ablation experiments area, the common for many ML models, rare but catastrophic prediction failures call for relation-aware methods (for example, graph ANN or physics-aware, or rather "kinematic-aware" ANN) that would limit sudden changes in the stock prediction in a limited time period. Attention-aware methods, which would look for similar input period models in the past, may also be for future research.

Such an approach is also methodologically co-located with the abovementioned limitations - the study of the "disruption information"-rich time intervals, perhaps in contrast with new information-deprived periods. For example, the study of the models trained immediately before and tested immediately after the "disruption information" boundaries, such as the emergence of the COVID pandemic or the start of the war in Ukraine. "Information" here is used in its common sense meaning rather than in the information theory's.

Because such information may have prolonged inertia to be digested by markets (for example, intelligence or other expert communities may have that information months in advance), an extended temporal horizon "cross-training" may be a prospective area of research.

Extension of the research to emergent markets and behaviour of the individual stocks, especially those which do not follow the expected trend (cross-training of the successful company stocks on the failing companies' data and vice versa), is another area of future research.

References

1. Beheim, L., Zitouni, A., Belloir, F., de la Housse, C.d.M.: New rbf neural network classifier with optimized hidden neurons number. WSEAS Transactions on Systems (2), 467–472 (2004)
2. Broomhead, D.S., Lowe, D.: Radial basis functions, multi-variable functional interpolation and adaptive networks. Tech. rep., Royal Signals and Radar Establishment Malvern (United Kingdom) (1988)
3. Eun, C.S., Shim, S.: International transmission of stock market movements. Journal of financial and quantitative Analysis **24**(2), 241–256 (1989)
4. Fama, E.F.: Efficient capital markets: A review of theory and empirical work. The journal of Finance **25**(2), 383–417 (1970)
5. Girosi, F., Poggio, T.: Representation properties of networks: Kolmogorov's theorem is irrelevant. Neural Computation **1**(4), 465–469 (1989)
6. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA (2001)
7. He, Q.Q., Pang, P.C.I., Si, Y.W.: Multi-source transfer learning with ensemble for financial time series forecasting. In: 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). pp. 227–233. IEEE (2020)
8. Ivakhnenko, A.G.: Polynomial theory of complex systems. IEEE transactions on Systems, Man, and Cybernetics (4), 364–378 (1971)
9. Jakaite, L., Schetinin, V., Maple, C., et al.: Bayesian assessment of newborn brain maturity from two-channel sleep electroencephalograms. Computational and Mathematical Methods in Medicine **2012** (2012)
10. Jensen, M.C.: The performance of mutual funds in the period 1945-1964. The Journal of finance **23**(2), 389–416 (1968)
11. Kolmogorov, A.N.: On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. American Mathematical Society (1961)
12. Kurkin, S.A., Pitsik, E.N., Musatov, V.Y., Runnova, A.E., Hramov, A.E.: Artificial neural networks as a tool for recognition of movements by electroencephalograms. In: ICINCO (1). pp. 176–181 (2018)
13. Kůrková, V.: Kolmogorov's Theorem Is Relevant. Neural Computation **3**(4), 617–622 (12 1991)
14. Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E.: Deep learning for stock market prediction. Entropy **22**(8), 840 (2020)
15. Nyah, N., Jakaite, L., Schetinin, V., Sant, P., Aggoun, A.: Evolving polynomial neural networks for detecting abnormal patterns. In: 2016 IEEE 8th international conference on intelligent systems (IS). pp. 74–80. IEEE (2016)

16. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation **3**(2), 246–257 (1991)
17. Pinkus, A.: Approximation theory of the mlp model in neural networks. Acta numerica **8**, 143–195 (1999)
18. Poterba, J.M., Summers, L.H.: Mean reversion in stock prices: Evidence and implications. Journal of financial economics **22**(1), 27–59 (1988)
19. Ren, S., Sun, J., He, K., Zhang, X.: Deep residual learning for image recognition. In: CVPR. vol. 2, p. 4 (2016)
20. Roll, R.: Orange juice and weather. The American Economic Review **74**(5), 861–880 (1984)
21. Roll, R.: What every cfo should know about scientific progress in financial economics: What is known and what remains to be resolved. Financial management **23**(2), 69–75 (1994)
22. Schetinin, V., Jakaite, L., Schult, J.: Informativeness of sleep cycle features in bayesian assessment of newborn electroencephalographic maturation. In: 2011 24th International Symposium on Computer-Based Medical Systems (CBMS). pp. 1–6. IEEE (2011)
23. Selitskaya, N., Sielicki, S., Jakaite, L., Schetinin, V., Evans, F., Conrad, M., Sant, P.: Deep learning for biometric face recognition: experimental study on benchmark data sets. Deep Biometrics pp. 71–97 (2020)
24. Selitskiy, S.: Kolmogorov's gate non-linearity as a step toward much smaller artificial neural networks. In: Proceedings of the 24th International Conference on Enterprise Information Systems. vol. 1, p. 492–499 (2022)
25. Selitskiy, S.: Elements of active continuous learning and uncertainty self-awareness: A narrow implementation for face and facial expression recognition. In: Goertzel, B.e.a. (ed.) Artificial General Intelligence. pp. 394–403. Springer (2023)
26. Selitskiy, S., Christou, N., Selitskaya, N.: Using statistical and artificial neural networks meta-learning approaches for uncertainty isolation in face recognition by the established convolutional models. In: Nicosia, G.e.a. (ed.) Machine Learning, Optimization, and Data Science. pp. 338–352. Springer (2022)
27. Selitsky, S.: Hybrid convolutional-multilayer perceptron artificial neural network for person recognition by high gamma eeg features. Medicinskiy Vestnik Severnogo Kavkaza **17**(2), 192–196 (2022)
28. Sewell, M.: History of the efficient market hypothesis. Rn **11**(04), 04 (2011)
29. Shleifer, A.: Inefficient markets: An introduction to behavioural finance. Oup Oxford (2000)
30. Stoean, C., Paja, W., Stoean, R., Sandita, A.: Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations. PloS one **14**(10), e0223593 (2019)
31. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
32. Tsang, G., Deng, J., Xie, X.: Recurrent neural networks for financial time-series modelling. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 892–897. IEEE (2018)
33. Venugopal, V., Baets, W.: Neural networks and statistical techniques in marketing research: A conceptual comparison. Marketing intelligence & planning (1994)
34. Wickstrøm, K., Kampffmeyer, M., Mikalsen, K.Ø., Jenssen, R.: Mixing up contrastive learning: Self-supervised representation learning for time series. Pattern Recognition Letters **155**, 54–61 (2022)

35. Yen, G., Lee, C.f.: Efficient market hypothesis (emh): Past, present and future. Review of Pacific Basin Financial Markets and Policies **11**(02), 305–329 (2008)
36. Zhang, G.P.: A neural network ensemble method with jittered training data for time series forecasting. Information Sciences **177**(23), 5329–5346 (2007)