Machine Learning for Earnings Prediction: A Nonlinear Tensor Approach for Data Integration and Completion

Ajim Uddin
New Jersey Institute of Technology
Newark, NJ, USA
au76@njit.edu

Xinyuan Tao
New Jersey Institute of Technology
Newark, NJ, USA
xtao@njit.edu

Chia-Ching Chou
Central Michigan University
Mount Pleasant, Michigan, USA
chou1c@cmich.edu

Dantong Yu
New Jersey Institute of Technology
Newark, NJ, USA
dtyu@njit.edu

ABSTRACT

Successful predictive models for financial applications often require harnessing complementary information from multiple datasets. Incorporating data from different sources into a single model can be challenging as they vary in structure, dimensions, quality, and completeness. Simply merging those datasets can cause redundancy, discrepancy, and information loss. This paper proposes a convolutional neural network-based nonlinear tensor coupling and completion framework (NLTCC) to combine heterogeneous datasets without compromising data quality. We demonstrate the effectiveness of NLTCC in solving a specific business problem - predicting firms' earnings from financial analysts' earnings forecast. First, we apply NLTCC to fuse firm characteristics and stock market information into the financial analysts' earnings forecasts data to impute missing values and improve data quality. Subsequently, we predict the next quarter's earnings based on the imputed data. The experiments reveal that the prediction error decreases by 65% compared with the benchmark analysts' consensus forecast. The long-short portfolio returns based on NLTCC outperform analysts' consensus forecast and the S&P-500 index from three-day up to two-month holding period. The prediction accuracy improvement is robust with different performance metrics and various industry sectors. Notably, it is more salient for the sectors with higher heterogeneity.

CCS CONCEPTS

• **Computing methodologies → Factorization methods**; **Learning latent representations**; **Neural networks**; • **Applied computing → Forecasting**; *Economics*; • **Information systems → Spatial-temporal systems**.

KEYWORDS

Sparse Tensor Completion, Nonlinear Tensor Factorization, Convolutional Neural Network, Firm Earnings Forecast, FinTech

ICAIF '22, November 2–4, 2022, New York, NY, USA

ACM ISBN 978-1-4503-9376-8/22/11…$15.00
https://doi.org/10.1145/3533271.3561677

ACM Reference Format:
Ajim Uddin, Xinyuan Tao, Chia-Ching Chou, and Dantong Yu. 2022. Machine Learning for Earnings Prediction: A Nonlinear Tensor Approach for Data Integration and Completion. In *3rd ACM International Conference on AI in Finance (ICAIF '22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3533271.3561677

1 INTRODUCTION

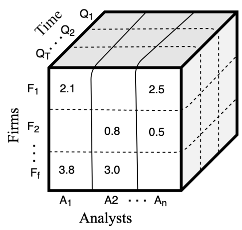
Finance studies often employ heterogeneous datasets from different sources with different structures. Some datasets are noisy, sparse, and unbalanced with missing values; some are unstructured, containing text or networks; some with high frequencies, intraday and daily, whereas others with low frequencies such as quarterly and annually. A simple combination of multiple datasets thus induces many challenges, including the curse of dimensionality, neglecting interactions among data attributes, and suffering significant information loss in aggregating data with different dimensions or frequencies. In addition, conventional econometric analyses, such as regressions, require input data to be complete and balanced, which is often not true in real-world applications.

To overcome those challenges, this paper proposes a NonLinear Tensor Coupling and Completion framework (NLTCC). NLTCC uses (sparse) tensor to represent input data, decomposes the input tensor into latent vectors to disentangle the complex multi-way relationships among input data, and then uses a convolutional neural network (CNN) to impute missing values and reconstruct the entire tensor. The key novelty of NLTCC is to perform nonlinear tensor factorization on multiple datasets simultaneously and extract low-rank representations. Compared to traditional linear tensor factorizations, i.e., CANDECOMP/PARAFAC (CP) [16] and Tucker decomposition [25] the neural network of NLTCC allows it to capture nonlinear interactions among datasets. As a result, NLTCC extracts more relevant information to impute missing values and mitigate the curse of dimensionality.

To investigate the advantage and the usefulness of the proposed NLTCC, we use financial analysts' forecast of earnings per share (EPS) data as the experiment for two reasons: its importance in the business world and its complex data structure. EPS is the ratio of a firm's earnings to its number of common shares outstanding. It reflects firms' performance and is one of the fundamental inputs for security pricing [21]. Investors are highly interested in firms' future EPS to making their investment decision. Analysts'

Quarter	Firms	A_1	A_2	A_n
Q_1	F_1	1.8		2.0
	F_2			1.5
	F_f	3.2	2.8	
Q_2	F_1	3.0		2.8
	F_2		0.5	0.4
	F_f	4	4.1	
Q_T	F_1	2.1		2.5
	F_2		0.8	0.5
	F_f	3.8	3.0	

(a) Matrix



(b) Tensor

Figure 1: Representing financial analysts earning forecast data in the matrix (a) and tensor formats (b).

consensus (mean or median) forecast of EPS is considered the most common measure of market expectation.[1] It serves as a benchmark for advanced yet complex prediction models and provides information to individual investors who do not have the necessary skills, knowledge, and time to conduct their own analysis. Studies in both finance and accounting show that analysts' earnings forecast is a better estimator of firm earnings than time series models because analysts incorporate their skills, experience, and timely information in their forecast [8, 15, 21].

Besides the importance, several challenges exist in harnessing analysts' earnings forecast data. First, analysts' EPS forecast is highly sparse and unbalanced. At a given time, multiple analysts follow one firm and generate individual reports. Analysts only track a limited set of firms and create reports for only these firms while skipping all other firms. In addition, analysts occasionally miss reporting and change the firms they follow. Resulting in a very high number of missing values in the dataset. Imputing these missing values can improve earning prediction [26, 27].

Second, the EPS dataset has three dimensions, *time*, *firms* and *analysts*. The common practice to ease the data complexity is aggregating information from individual analysts to the firm-level and flattening the third-order tensor into a *firm-time* matrix, as shown in Figure 1a [5, 6]. However, time-varying complex relationships exist among analysts, firms, and industries [4, 12]. The *firm-time* matrix only preserves information from the time domain and neglects inter-firm relationships in the spatial domain [18]. Besides, analysts vary in their forecasting accuracy [7, 23] and may show a systematic bias [7, 12]. Aggregating all the efficient and inefficient analysts' forecasts indiscriminately will contaminate the information content of efficient analysts' forecasts. NLTCC solve this challenge by representing panel data in their natural third-order tensor form (Figure 1b) to capture the inter-dependency of spatial and temporal domains and learn separate embedding for each analyst.

Third, EPS announcement comes quarterly or lower frequency along with individual forecasts of hundreds of analysts, resulting in more predictors than observations. Meanwhile, EPS can be affected and captured by other higher frequency factors, such as firm characteristics and market returns. NLTCC proposes a novel method to aggregate these higher frequency data with analysts' forecast data to incorporate more timely information in return prediction.

[1] We define the "consensus prediction" as the average of all available analysts forecast for a firm at a given quarter and refer it to as "mean prediction".

We show the superiority of NLTCC from three perspectives. First, we use NLTCC to impute missing values in individual analysts' forecasts with complementary data, i.e., firm characteristics and daily stock returns. NLTCC reduces the error of missing value imputation by 57% compared to the standard matrix factorization (MF) [20] and by 48% compared to the linear tensor factorization algorithm (CPWOPT) [1]. Second, with the high-quality imputed data, we apply ML to predict firms' next quarter's earnings. Results show that NLTCC improves the prediction accuracy by 5% for R^2 and 6% for MAPE compared with analysts' consensus forecasts. Finally, to evaluate the economic significance, we construct a long-short portfolio based on NLTCC predictions. The Sharpe ratio of the long-short portfolio for a three-day holding period is 2.68, an improvement by 2.03 from the S&P-500 index Sharpe ratio.

The main contributions of our work are as follows:

- We propose a nonlinear tensor framework to address the challenges of heterogeneous data integration, missing values imputation, and different data frequencies. NLTCC provides a more practical approach for researchers and industry practitioners to work with multiple datasets with different structures.
- We demonstrate that embedding learning with CNN and spatial-temporal regularization can harness maximum information from heterogeneous data without compromising quality. By enforcing orthogonal regularization on temporal dimension and local similarity on spatial dimension, we ensure high-quality embedding for data imputation and downstream prediction.
- Our works contribute to the earnings prediction literature in finance by demonstrating the importance of capturing latent interactions among firms and analysts to improve prediction accuracy.

The rest of the paper is organized as follows: Section 2 briefly introduces tensor factorization. Section 3 presents the details of the NLTCC model. Section 4 discusses the data, experimental setup, and hyperparameters used for this study. Section 5 presents the experimental results and discusses the economic significance of the proposed model. Section 6 provides sensitivity analysis regarding regularization and ranks and discusses the computational complexity of the model. Section 7 offers conclusions.

2 BACKGROUND

Tensor factorization/decomposition is the low-rank approximation for higher-order data. Tensor decomposition has become a prevalent dimensionality reduction technique in signal processing, computer vision, and graph analysis [19, 24]. CP is one of the most popular low-rank tensor factorization models [9, 16]. CP factorizes a tensor into a series of rank-one tensors and approximates the original tensor with the sum of the r rank-one component tensors. For a 3rd-order tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and a given rank r, CP factorization essentially consists of three-factor matrices $U \in \mathbb{R}^{d_1 \times r}$, $V \in \mathbb{R}^{d_2 \times r}$ and $W \in \mathbb{R}^{d_3 \times r}$ and is expressed as:

$$\mathcal{X} \approx [\![U, V, W]\!] \equiv \sum_{s=1}^{r} u_s \circ v_s \circ w_s$$

Tensor completion first applies factorization on a partially observed tensor $\mathcal{X}$ to learn low-rank factor-matrices from the observed entries and then impute missing entries to complete the target tensor $\hat{\mathcal{X}}$ from the factor matrices. We define the element-wise CP reconstruction as follows:

$$\hat{\mathcal{X}}_{i,j,k} = \sum_{s=1}^{r} U_{si} V_{sj} W_{sk} \tag{1}$$

The objective function tries to minimize the following equation:

$$f(U, V, W) = \|(\mathcal{X} - \hat{\mathcal{X}}) \odot \mathbb{1}_{\mathcal{X}}\|_F^2$$

Coupled tensor factorization factorizes two tensors concurrently. The underlying assumption is that two Nth-order tensors $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_n}$ and $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_n}$ share at least one common dimension. Coupled tensor algorithms propagate the information from one tensor to the other by enforcing the same latent factor matrices for the shared dimensions during the factorization process [2, 3]. For two third-order tensors $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ with one common dimension d_1, the objective function for coupled tensor is to minimize the mean square error of two tensor factorizations as follows:

$$f(U, V, W, Q, T) = \left|\mathcal{X} - [[U, V, W]]\right|_F^2 + \lambda \left|\mathcal{Y} - [[U, Q, T]]\right|_F^2$$

Here, λ is the hyper-parameter to adjust the relative importance between the two tensors. The low-rank factor matrices are U, V, W, T, Q with U shared by both tensors. Similarly, the coupled tensor completion imputes a missing value in tensor $\mathcal{X}$ as $\hat{\mathcal{X}}_{i,j,k} = \sum_{s=1}^{r} U_{si} V_{sj} W_{sk}$ and in tensor $\mathcal{Y}$ as $\hat{\mathcal{Y}}_{i,j,k} = \sum_{s=1}^{r} U_{si} Q_{sj} T_{sk}$, where r is the tensor rank.

3 METHODOLOGY

3.1 The Model Architecture

The architecture of NLTCC present in Figure 2 consists of four modules: a data fusion module, an embedding module, a nonlinear mapping module, and an aggregation module, each of which implements one step in the forward propagation.

Data fusion module: The data fusion module takes three different tensors as input: firm characteristics $C \in \mathbb{R}^{q \times f \times c}$, analysts' EPS forecasts $\mathcal{A} \in \mathbb{R}^{q \times f \times a}$, and firm daily return $\mathcal{R} \in \mathbb{R}^{f \times d'}$. One order – the number of firms (f) – is the same across the three tensors. The firm characteristics and analyst EPS forecast have the quarterly (q) dimension (the first order). The return data has a daily frequency, but by folding the two-dimensional return tensor ($\mathcal{R} \in \mathbb{R}^{f \times d'}$) on the second-order ($d'$), we convert it into a three-dimensional tensor ($\mathcal{R} \in \mathbb{R}^{q \times f \times d}$), whereas $d = d'/q$. We partition the daily returns in quarters and represent the days within a quarter as features of that quarter, resulting in a third-order ($quarter \times firm \times day$) tensor. Such folding operation makes the first two dimensions (firm and time) of all three tensors identical and can then be concatenated along the third dimension of tensors.

$$\mathcal{X} = \mathcal{A} \| C \| \mathcal{R} \tag{2}$$

where $\mathcal{X} \in \mathbb{R}^{q \times f \times z}$, $z = characteristics + analysts + daily\ returns$, and $\|$ the concatenation operation. We treat the characteristics, returns, and analysts' earnings forecast in the concatenated tensor as the firm's unified features. By doing so, we ensure the subsequent tensor completion uses complete knowledge from both daily return and characteristics data to augment the imputation procedure performed on the analysts' forecast data.

Embedding module: The embedding module use the fused third-order tensor $\mathcal{X}$ as training data to learn three factor matrices $U \in \mathbb{R}^{q \times r}, V \in \mathbb{R}^{f \times r}$ and $W \in \mathbb{R}^{z \times r}$ as latent embeddings. Instead of performing a simple linear aggregation as in Equation 1, we adopt the non-linear aggregation method proposed in CoSTCo [22], and design the neural networks that essentially implement the following parameterized function on the domain of tensor space, and map the indices of a tensor cell and associated embedding to the corresponding tensor element $\hat{\mathcal{X}}_{i,j,k}$:

$$\hat{\mathcal{X}}_{i,j,k} = f(i, j, k) = f(U_{i:}, V_{j:}, W_{k:}, \{\theta_i, \ldots, \theta_n\}) \tag{3}$$

where $1 \le i \le d_1$, $1 \le j \le d_2$, $1 \le k \le d_3$ and $\theta_i, \ldots, \theta_n$ are the weights of convolutional layers, dense layers, and regularization, respectively. Equation 3 defines an element-wise tensor completion based on the embedding matrices and the neuron weights. During network training, given an entry index i, j, k, the neural network first forward-propagates to obtain the function value and then back-propagates the loss to the embedding layers for updating the elements in U, V, W.

Nonlinear mapping module: In this module, NLTCC uses convolutional neural network to perform element-wise tensor reconstruction and completion for the sparse input tensor. Liu et al. (2019) prove that convolutional layers are more efficient in terms of the number of parameters than MLP, especially for learning high-quality nonlinear embeddings in factor matrices [22]. The subsequent reconstruction uses the embeddings to estimate a sparse tensor's unobserved entries with higher accuracy than other available linear/nonlinear tensor completion methods. The nonlinear mapping module uses two 2-D convolutional layers with filter size of $(1, 3)$ and $(r, 1)$. The output of each convolutional layer is:

$$\begin{aligned} \mathcal{H}^1_{conv} &= \sigma(Conv(\mathcal{H}_{env} : (1, 3))) \in \mathbb{R}^{C \times 1 \times 3} \\ \mathcal{H}^2_{conv} &= \sigma(Conv(\mathcal{H}^1_{conv} : (r, 1))) \in \mathbb{R}^{C \times 1 \times 1} \end{aligned} \tag{4}$$

where C is the channel number and $\sigma(\cdot)$ is the nonlinear activation function ReLU $\sigma = max(\cdot, 0)$.

Aggregation module: The aggregation module first takes the second convolutional layer $\mathcal{H}^2_{conv}$'s output, flattens it into a length-C vector, and uses fully connected network layers to aggregate the vector into a scalar as the reconstructed entry i, j, k in the output tensor. $\hat{\mathcal{X}}$.

3.2 Objective Function and Regularization

We first define the objective to minimize the mean squared loss between the observations in $\mathcal{X}$ and the reconstructed values in $\hat{\mathcal{X}}$. Then we back-propagate the gradient of the objective function to the embedding layer and update all affected parameters in $U_{i,:}$, $V_{j,:}$, and $W_{k,:}$. We define the objective function as follows:

$$\mathcal{L} = \min_{U,V,W} \|(\hat{\mathcal{X}} - \mathcal{X}) \odot \mathbb{1}_{\mathcal{X}}\|_F^2 + \mathcal{R}(U, V, W, \theta) \tag{5}$$

$$= \min_{U,V,W} \sum_{(i,j,k,y) \in s} (f(i, j, k) - \mathcal{X}_{i,j,k})^2 + \mathcal{R}(U, V, W, \theta) \tag{6}$$

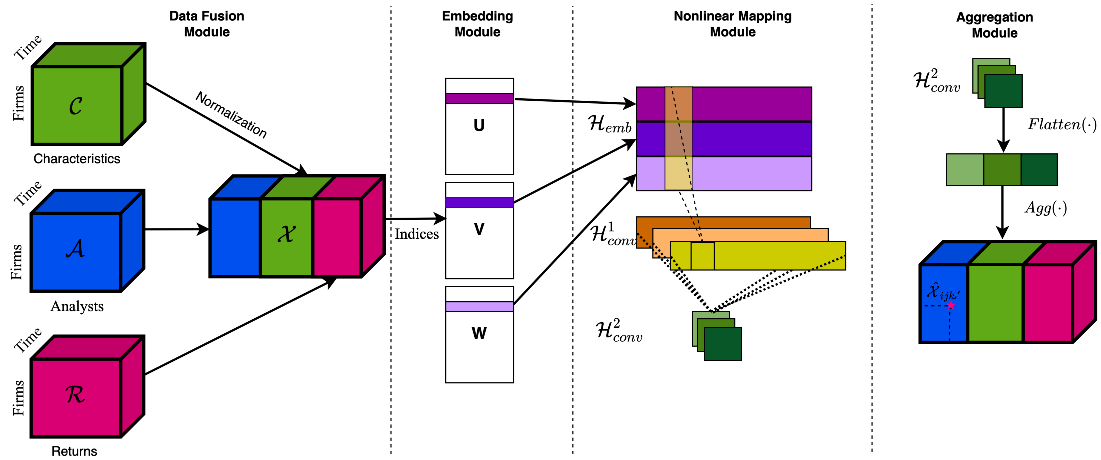


Figure 2: Model architecture

Here $s = \{(i, j, k, y_{ijk})|\mathbb{1}_{ijk} = 1\}$ represents the training and testing data. The Hadamard product $(\hat{\mathcal{X}} - \mathcal{X}) \odot \mathbb{1}_{\mathcal{X}}$ is applied to select the observed values because the loss function only considers non-missing values and ignores the imputed values that have no ground truth for comparison.

Regularization to enforce orthogonality: Concatenating multiple tensors allows useful information to propagate from one tensor to another. However, the concatenation might also introduce noises, inconsistency, and redundancy with information. Therefore, explicit regularization is required to impose penalties on such unwanted information. Here, we impose the orthogonality constraints on the time dimension to eliminate the redundancy among learned features and associated undesired artifacts (high sensitivity and high variance) in the downstream learning tasks. The orthogonal regularization ensures that $U_{:i} \perp U_{:j}$ when $U_{:i} \neq U_{:j}$. It also ensures that all individual features in the embedding matrix are as different as possible. The new objective function with orthogonality constraint is defined as follows:

$$\sum \|(\mathcal{X} - \hat{\mathcal{X}}) \odot \mathbb{1}_{\mathcal{X}}\|_2 + \lambda_1 \|U^\top U \odot (\mathbf{1}\mathbf{1}^\top - I)\|_{1,1} \tag{7}$$

where $\mathbf{1}\mathbf{1}^\top$ is the matrix of all ones. It penalizes the inter-dependency among the latent features, i.e., the column vectors $U_{:i}$ and $U_{:j}$ in the quarter matrix U. As a result, the embedding features in U are as distinct as possible. The rationale behind this is to capture as much significant information as possible from the temporal dimensions without inducing collinearity among features.

Regularization to enforce return similarity and locality: High-quality embedding must preserve the internal data structure and locality of the objects to be represented. In this case, the embeddings for closely related firms must be similar while different from those unrelated firms. We use the similarity regularization in the embedding matrix to enforce this condition. The correlation between two firms' returns is a strong indicator of similarity and might be closely related to the explicit characteristics of firms (industry sector, firm size, revenues, and markets) and latent features to be learned during tensor completion. To build the similarity matrix, we compute the pairwise correlation between firms and build the cut-similarity matrix $S_{m,n}$ using a threshold function $\zeta = 0.30$. The threshold function makes S sparse by eliminating both trivial and negative correlations. The normalized cut similarity matrix $S \longleftarrow D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$ with diagonal degree matrix of D where $d_n = \sum_{l=1}^{N} S_{nl}$ enhance the total similarity between the related firms and dissimilarity between unrelated firms in the firm embedding matrix. The new objective function with the orthogonal regularization on the time dimension and similarity on the firm dimension is defined as follows:

$$\sum \|(\mathcal{X} - \hat{\mathcal{X}}) \odot \mathbb{1}_{\mathcal{X}}\|_2 + \lambda_1 \|U^\top U \odot (\mathbf{1}\mathbf{1}^\top - I)\|_{1,1} + \lambda_2 \|S - VV^T\|_F^2 \tag{8}$$

4 DATA AND EXPERIMENTAL DETAILS

4.1 Data

There are three primary sources for the data: quarterly EPS and individual analysts' forecast data from Thomson Reuters I/B/E/S, firm characteristics from COMPUSTAT, and the stock information from CRSP. These data are available at https://wrds-www.wharton.upenn.edu/. We consider the period from Q1-2009 to Q4-2017.

To compare the performance of NLTCC with other benchmarks, we tested all models on a combined sub-sample of 300 firms with 173 analysts following. In addition, we also analyzed the model performance in firm groups according to their industry sectors. In data cleaning, we remove firms with time lapses in the complete time series and analysts who have predicted for less than four years or made less than 200 predictions in their entire career. This allows us to have consistent data sets for sound performance evaluation. Table 1 shows data description for the combined group, different industry sectors, and sub-sectors within the manufacturing sector (bottom panel), including the number of firms (Firms), the number of analysts (Analysts), missing percentage of third-order tensors, the volatility of individual analyst's forecast (Analysts STD), and the volatility of realized EPS (EPS STD). To avoid the size effect, we scale the two standard deviations by their respective means. For characteristics data we follow the common literature [5, 14, 17, 27] to select nineteen most influential firms characteristics.[2]

[2]The list of these characteristics and their calculation method is available in [27].

Table 1: Data Description

Industry Class	Firms	Analysts	Missing (%)	Analysts STD	EPS STD
Combined	300	173	96.65	2.8471	0.9232
Mining	57	152	88.36	3.6396	3.2796
Construction	18	81	87.29	2.0617	0.9084
Manufacturing	306	170	97.03	2.0969	0.8335
Transportation	85	62	88.66	3.8768	1.0187
Wholesale trade	21	20	78.41	2.2829	0.4074
Retail trade	84	119	90.07	3.2147	1.4486
Finance	192	159	95.50	4.2137	1.9697
Service	122	101	93.68	2.4531	0.9849
Manufacturing Sub-Sector					
Food	21	55	81.38	3.1452	1.0493
Textile	32	75	91.28	2.6940	0.9587
Papers	12	29	73.49	2.3089	1.9054
Chemicals	57	146	94.55	2.7495	1.6615
Glass and metals	60	140	94.08	2.5267	0.7339
Computers	72	89	93.22	3.0194	1.4772
Automobile	56	86	92.75	2.2095	1.2185

Table 2: Hyperparameter Settings

Model	Hyperparameter	Values
MF	Rank (r)	5, 10, 15, 20
	Learning Rate (e)	1e-5, 1e-4, **1e-3**, 1e-2
	Regularization (λ)	1e-4, **1e-3**, 1e-2, 1e-1
CPWOPT	Rank (r)	5, 10, 15, 20
NLTCC	Rank (K)	5, **10**, 15, 20
	Learning Rate (e)	1e-5, **1e-4**, 1e-3, 1e-2
	Regularization (λ)	1e-4, **1e-3**, 1e-2, 1e-1
SVR	Epsilon (ϵ)	1e-5, **1e-4**, 1e-3, 1e-2
	Constant (C)	1, 2, 3, **5**, 10
XGBoost	Learning Rate (e)	1e-5, 1e-4, 1e-3, **1e-2**
	Regularization (λ)	1e-4, 1e-3, **1e-2**, 1e-1

4.2 Experimental Setup and Hyperparameters

The NLTCC model is implemented using Keras [11] with the TensorFlow at the back-end.[3] To overcome the information leakage problem, following [27], we conduct our experiments based on the rolling window with a window size of 28. First, we perform the tensor imputation in the first 28 quarters and then train the prediction model in the first 27 quarters and test the model performance only in the last quarter. We then iteratively move the rolling window forward by one quarter, impute data again, and then perform with the same training and test split. Finally, we report the average value of the test quarters.

We initially tune the hyperparameters (reported in Table 2) using a grid search on 10% validation data for the first window of the rolling window process, i.e., Q1-2009 to Q4-2015. The best parameter set (bold front) is then used for the remaining windows. For the neural network, the batch size is 128. The maximum training Epochs is set to 500 with the early stopping criteria if the validation loss stops decreasing for ten Epochs.

[3]The python codes for NLTCC with sample data are publicly available at https://github.com/Ajim63/NLTCC.git.

5 RESULTS AND DISCUSSIONS

We evaluate NLTCC against other existing models in three evaluation criteria. First, accuracy in imputing missing values in analysts' forecast data with and without complementary datasets. Second, accuracy in predicting firms' next quarter earnings. Third, improvement in portfolio return and Sharpe ratio.

5.1 Tensor Completion

We compare the imputation performance of NLTCC against two well-known missing value imputation techniques; MF [20], and CP-based tensor completion CPWOPT [1]. To evaluate the algorithms' robustness, we first randomly sample the available EPS forecast data by 90%, 80%, 60%, and 40% to create a series of tensors with the additional missing values from an already sparse tensor and then evaluate data imputation on these tensors with an increasing level of sparsity. The performance is evaluated in terms of mean squared error (MSE) and tensor completion score (TCS)[1].

The tensor completion results are presented in Table 3. Both MF and CPWOPT are for single tensor completion. Therefore, to get a fair comparison, we analyze two versions of our model. NLTCC(A) is the nonlinear tensor completion with only analysts' forecast data, whereas NLTCC further incorporates firm characteristics and stock market information. All three tensor completion methods outperform the matrix completion by a significant margin. Table 3 shows that the performance difference among these models is minimal at 10% additional missing value. As the percentage of missing values increases, the superior performance of NLTCC becomes highly evident in both TCS and MSE. When 98% entries are missing, at rank 10, NLTCC outperforms MF by 57%, and CPWOPT by 48%. Even with only a single dataset, our approach NLTCC(A) outperforms MF and CPWOPT by 32% and 17%, respectively. The importance of using auxiliary information is visible in the performance difference between NLTCC(A) and NLTCC. Fusing two additional datasets: firm characteristics and stock return, further improves tensor completion accuracy by 36%.

5.2 Predicting Firm Earnings

We compare the prediction accuracy of imputed data using NLTCC against those imputed by matrix factorization and CPWOPT. Two prediction methods, Support Vector Regression (SVR) [13] and XGBoost [10], are applied in predicting future earnings. As XGBoost has built-in sparsity awareness and handles missing values in datasets, we also use XGBoost to predict earnings from the original dataset with missing values directly. XGBoost allows us to compare the performance improvement with/without our tensor completion for data imputation. We also directly compare our prediction model with the industry benchmark mean prediction. For evaluating model performance, we use three performance metrics, R^2, MSE, and MAPE.

Table 4 shows that both NLTCC+SVR and NLTCC+XGBoost outperform the mean prediction in all three performance measurements. For example, NLTCC+XGBoost exceeds the mean prediction in R^2 by 5%, in MSE by 65%, and in MAPE by 6%. Meanwhile, the other two imputation techniques MF and CPWOPT can not beat the mean prediction due to those models' inherent limitations. Particularly, to impute data, MF collapses three-dimensional tensor into

Table 3: Tensor Completion Results

Additional Missing		10%			20%			40%			60%		
Total Missing		96.09%			96.52%			97.39%			98.26%		
Metric	Model/Rank	5	10	20	5	10	20	5	10	20	5	10	20
TCS	MF	0.0689	0.0553	0.0522	0.1680	0.1538	0.1532	0.3244	0.3322	0.3351	0.7575	0.7964	0.7815
	CPWOPT	**0.0495**	0.0315	0.0303	0.1371	0.1259	0.1592	0.3752	0.2579	0.4079	0.8353	0.6543	0.8942
	NLTCC(A)	0.0723	0.0430	0.0302	0.1462	0.1209	0.1120	0.2853	0.2572	0.2158	0.5702	0.5410	0.5143
	NLTCC	0.0646	**0.0292**	**0.0289**	**0.1362**	**0.1102**	**0.1009**	**0.2653**	**0.2002**	**0.1914**	**0.4011**	**0.3422**	**0.3212**
MSE	MF	0.0113	0.0092	0.0082	0.0280	0.0209	0.0208	0.0576	0.0580	0.0556	0.2067	0.1775	0.1774
	CPWOPT	**0.0107**	0.0062	0.0050	0.0156	0.0736	0.0559	0.3733	0.3573	0.6573	0.2833	0.8265	0.8308
	NLTCC(A)	0.0122	0.0104	0.0042	0.0181	0.0188	0.0106	0.0461	0.0359	0.0252	0.0769	0.0570	0.0436
	NLTCC	0.0109	**0.0056**	**0.0039**	**0.0161**	**0.0172**	**0.0101**	**0.0381**	**0.0309**	**0.0212**	**0.0569**	**0.0270**	**0.0236**

Table 4: Predicting Firms Earnings

A: Performance Evaluation			
Model	R^2	MSE	MAPE(%)
Mean	0.9320	0.0498	55.1690
XGBoost	0.8712	0.0502	71.5423
MF+SVR	0.8451	0.0611	64.0114
MF+XGBoost	0.8526	0.0521	62.3044
CPWOPT+SVR	0.7616	0.0819	94.5147
CPWOPT+XGBoost	0.8133	0.0749	89.8420
NLTCC+SVR	0.9665	0.0195	54.1246
NLTCC+XGBoost	**0.9765**	**0.0172**	**51.6342**

B: Mean Difference Test of R^2	
Model	Difference
NLTCC+XGBoost - Mean	0.0445**
NLTCC+XGBoost - XGBoost	0.1053***

Note: **, ***, significant at 5%, and 1% level of significance.

Table 5: Earnings Prediction for Industry Groups

Industry	Model	R^2	MSE	MAPE
Mining	Mean	0.8061	0.1306	103.5411
	XGBoost	0.7527	0.1578	137.0901
	NLTCC+SVR	**0.8732**	**0.0509**	51.5701
	NLTCC+XGBoost	0.8660	0.0689	**25.1641**
Construction	Mean	0.9297	0.6754	54.5707
	XGBoost	0.9482	0.5560	57.0514
	NLTCC+SVR	0.9519	0.5265	49.3445
	NLTCC+XGBoost	**0.9692**	**0.4015**	**48.5945**
Manufacturing	Mean	**0.8927**	**0.0689**	**24.4744**
	XGBoost	0.6977	0.1888	106.6645
	NLTCC+SVR	0.8918	0.0922	32.5341
	NLTCC+XGBoost	0.8854	0.1023	39.6142
Transportation and public utilities	Mean	0.7973	0.0845	24.3018
	XGBoost	0.8116	0.0666	29.1281
	NLTCCC+SVR	0.8954	0.0589	13.5846
	NLTCCC+XGBoost	**0.9352**	**0.0501**	**13.5108**
Wholesale trade	Mean	**0.9468**	**0.0111**	**13.9946**
	XGBoost	0.9329	0.0268	18.7945
	NLTCC+SVR	0.8783	0.0355	21.3545
	NLTCC+XGBoost	0.8919	0.0370	23.4045
Retail trade	Mean	0.8939	0.2120	62.3345
	XGBoost	0.8538	0.2143	69.5145
	NLTCC+SVR	0.9298	0.1944	57.3745
	NLTCC+XGBoost	**0.9465**	**0.1675**	**55.5448**
Finance, insurance and real estate	Mean	0.5221	0.4799	24.9981
	XGBoost	0.4168	0.6443	59.2015
	NLTCC+SVR	0.6998	0.3315	23.8848
	NLTCC+XGBoost	**0.7206**	**0.3293**	**20.4585**
Service	Mean	0.9005	0.1128	52.2945
	XGBoost	0.9134	0.0986	55.6301
	NLTCC+SVR	0.8935	0.1203	41.2801
	NLTCC+XGBoost	**0.9323**	**0.0951**	**40.4948**

two dimensions by flattening time and firm into one dimension. Consequently, it fails to capture important information in the temporal dimension. Besides, firm-specific latent information is also lost when the data is flattened across firms into a matrix. Similarly, CPWOPT is a linear model and fails to capture nonlinear interactions among analysts and firms. The advantage of imputation with NLTCC and data fusion is evident in the performance comparison between XGBoost and NLTCC+XGBoost. XGBoost, with its internal data imputation alone, can not outperform the simple mean. Nevertheless, with NLTCC based imputed data, the same machine learning model with similar hyper-parameters increases the prediction performance by almost 12% (R^2). In addition, NLTCC is the only model better than the mean prediction. This suggests that the advantage of a prediction model is conditional on the quality of input data.

5.3 Robustness Test with Industry Sectors and Sub-Sectors

To check the robustness of our findings we group firms into their related industry sectors following the Standard Industrial Classification (SIC) codes and perform industry-wise tensor completion. Table 5 reports the result of predicting the next quarter earnings for firms in each industry group and Table 6 report one industry (manufacturing) at at a micro-level. There are several interesting findings.

First, the performance of NLTCC is relatively stable and consistent across groups. Except for the manufacturing and wholesale trade groups, the imputation based models (NLTCC+SVR and

Table 6: Earnings Prediction in Manufacturing Sub-Sectors

Industry	Model	R^2	MSE	MAPE
Food	Mean	0.8585	0.0471	49.9463
	XGBoost	0.8979	0.0435	50.7263
	NLTCC+SVR	**0.9243**	**0.0237**	47.7413
	NLTCC+XGBoost	0.8719	0.0473	**42.1763**
Textile	Mean	0.8613	0.0483	66.1538
	XGBoost	0.8330	0.0443	61.9275
	NLTCC+SVR	**0.8887**	**0.0225**	**51.9138**
	NLTCC+XGBoost	0.8513	0.0410	65.4938
Paper	Mean	0.8570	0.0132	21.7800
	XGBoost	0.8189	0.0202	18.4225
	NLTCC+SVR	0.8746	**0.0100**	18.7638
	NLTCC+XGBoost	**0.8760**	0.0101	**17.3750**
Chemicals	Mean	0.8204	0.7047	0.6498
	XGBoost	0.8157	0.2947	0.7465
	NLTCC+SVR	**0.8601**	**0.1502**	**0.5068**
	NLTCC+XGBoost	0.8540	0.1598	0.5221
Glass and metals	Mean	**0.9296**	**0.0219**	**35.5488**
	XGBoost	0.8041	0.0615	80.6163
	NLTCCC+SVR	0.9038	0.0420	59.9600
	NLTCCC+XGBoost	0.8982	0.0424	58.4575
Computers and electronics	Mean	0.9062	0.0389	30.5213
	XGBoost	0.8756	0.0350	52.0375
	NLTCC+SVR	0.9137	0.0318	28.5525
	NLTCC+XGBoost	**0.9283**	**0.0279**	**26.2750**
Automobile and air-crafts	Mean	0.9078	0.0536	29.0275
	XGBoost	0.8517	0.0754	44.7750
	NLTCC+SVR	**0.9386**	**0.0472**	**24.7925**
	NLTCC+XGBoost	0.9259	0.0512	26.5388

NLTCC+XGBoost) outperform the mean and simple XGBoost predictions in all three performance metrics for all other groups. Second, the performance improvement of NLTCC over the mean prediction is more prominent for groups with high analysts' dispersion (Analysts STD). For example, the analysts' dispersion of the mining (3.64), transportation (3.87), finance (4.21), and food (3.14) have highest Analysts STD and demonstrate highest performance improvement. In terms of R^2, the gain is 8% for mining, 17% for transportation, 38% for finance, and 8% for food. Third, the performance improvement of tensor completion is positively correlated to firms' realized EPS volatility. For firms with less variation in their actual earnings, the mean algorithm attains almost perfect prediction. Forth, with the high-quality data preprocessed by NLTCC imputation, two prediction models (NLTCC+XGBoost and NLTCC+SVR) have a negligible performance difference. The finding confirms that data preprocessing based on data fusion and imputation plays a vital role in data analysis, reveals the majority of knowledge from data, and eases the selection of downstream prediction models.

The performance improvement of CNN-based NLTCC over other methods can be associated with its ability to generate meaningful embedding vectors for the respective dimensions, i.e., quarter, firm, and analyst. Figure 3 provides a useful demonstration of this. We use the t-distributed stochastic neighbor embedding (t-SNE) to visualize our learned latent factor and their embedding space. For

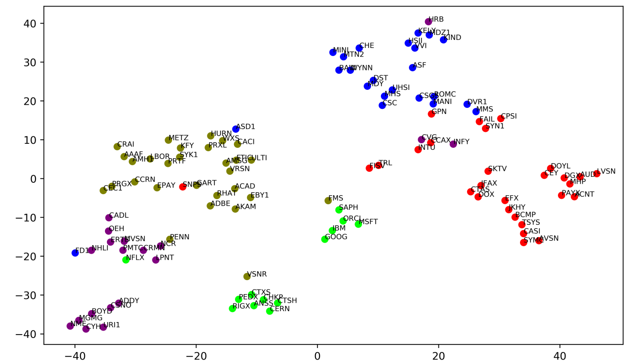


Figure 3: Spectral clustering based on firm latent factors. All the firms in the service industry are grouped into five clusters. The adopted tensor completion model learns meaningful embeddings for firms according to their size, service type, and the client groups they serve. Both color and space represent differences in embedding.

brevity, we use the clustering result of the service sector as an example. NLTCC learns multi-dimensional embedding for firms based on their size, service type, and the client groups they serve. For example, Citrix Systems, ANSYS, Check Point Software Technologies are IT companies and belong to the same group as Microsoft, Google, IBM, Oracle, and SAP, but their embedding space in the lower green group is slightly different from the large IT firms in the upper green group. High-quality embedding offers better grouping and classification accuracy and provides useful information about firms' inherent structure and hidden representations that are otherwise difficult to capture with simple data mining techniques.

Finally, the differences in performance improvement across analysts' dispersion and EPS volatility can be explained with the central limit theorem and confidence interval. For low-volatile firms, surpassing the simple yet effective mean prediction is challenging because when the data volatility is low, the mean prediction has a narrow confidence interval to encompass the actual earnings. In contrast, firms with insufficient or complicated information will have high analyst dispersion as analysts have different interpretations of the existing information. Under these circumstances, our NLTCC model help to select efficient information from analysts without introducing too much noise. As a result, it improves the quality of imputed data and further enhances prediction accuracy by machine-learning techniques.

5.4 Portfolio Analysis

To evaluate the proposed NLTCC's economic feasibility, we construct a long-short portfolio based on the difference between NLTCC predicted EPS and the mean consensus forecast. One day before each announcement day (t-1), we categorize stocks into "winner", "loser", and the rest, based on the difference between NLTCC prediction and mean forecast scaled by the share price at t-1. If the scaled difference of a stock is higher than 0.5%, it is considered a "winner", and if the difference is less than −0.5%, a "loser". The daily

Table 7: Long-Short Portfolio Performance

Holding periods	S&P-500	3 days	5 days	10 days	21 days	42 days
Average (%)	0.0451	0.3893	0.3334	0.2049	0.0845	0.0459
STD (%)	1.0996	2.3019	1.5496	0.8790	0.4245	0.2980
Sharpe Ratio	0.6507	2.6846	3.4159	3.7015	3.1602	2.4470

Table 8: Long Side and Short Side Comparison

Holding periods	3 days		5 days		10 days		21 days	
	Long	Short	Long	Short	Long	Short	Long	Short
Average	0.2939	0.1046	0.2676	0.0891	0.2063	0.0513	0.1375	-0.0095
STD	1.6566	2.5759	1.2159	1.9267	0.7978	1.2063	0.4749	0.7233
Sharpe Ratio	2.8158	0.6444	3.4934	0.7345	4.1041	0.6755	4.5968	-0.2090

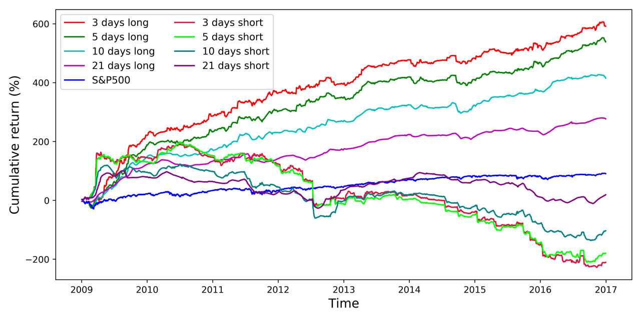


Figure 4: Cumulative returns on long-short portfolio.

return for each stock is calculated as:

$$r_{it} = \begin{cases} log(P_t + dividend) - log(P_{t-1}), & \text{if } \Delta > 0.005 \\ log(P_{t-1}) - log(P_t + dividend), & \text{if } \Delta < -0.005 \\ NA, & \text{otherwise.} \end{cases} \quad (9)$$

where, $\Delta = \frac{NLTCC_{iq} - Mean_{iq}}{P_{t-1}}$ is the scaled difference of each stock. We initially hold the S&P500 index ETF until appropriate "winner" ("loser") stocks are identified, and then take long (short) positions on selected stocks during the designated holding window around earnings announcement dates. At the end of the assigned holding window, we will cash out the positions, reinvest on S&P-500 and repeat the same process. We calculate the portfolio return as follows:

$$PR_t = \begin{cases} r_{S\&P,t} & \text{if } \forall r_{it} = NA \\ \frac{\sum_i^p r_{it}}{n}, & \text{otherwise.} \end{cases}$$

where, p is the number of stocks in the "winner" or "loser" group based on Equation 9. If multiple stocks meet our investment design for any given day, we hold an equally weighted position across all of them. Whenever a new stock is added, the old portfolio's weights are re-balanced to accommodate the new stock(s) with the same weights for all stocks.

Table 7 reports the results of the long-short portfolio for three-, five-, ten-, 21-, and 42-day holding period, respectively. Our hybrid portfolios for the five holding periods generate much higher returns than simply holding S&P-500 index ETF. The result also demonstrates that the majority of the above-average return comes from the information advantage on and right after the announcement day. Holding stocks for a longer period does not necessarily provide more additional benefits. Instead, the return for long holding periods shows a mean reversal trend. For 42 days holding period, the portfolio return is almost the same as S&P-500.

Table 8 reports the long side and short side of the portfolio separately. It is evident that the long side of the portfolio comes with a higher return with lower variance than the short side of the portfolio. Using a three-day holding period as an example, the long side of the portfolio earns almost three times as the short side. For the twenty-one day holding period, the short side return becomes even negative. Figure 4 shows that, from the beginning of 2009 to almost the end of 2012, the short portfolio return is above zero, indicating that we receive a loss in short-selling. This is intuitive: during this period, as the market recovered fast, the stock price had a significant upward trend. Nevertheless, over time, both the short and long positions of the portfolio become profitable and earn significantly higher returns than S&P-500.

6 ABLATION STUDY

For ablation study we evaluate several versions of NLTCC on the 300 firms analyzed in Section 5.2. These models include only financial analysts data "NLTCC(A)", financial analysts and firm characteristics data "NLTCC(A+F)", financial analysts, firm characteristics, and return data with no regularization "No_reg", all three datasets with the orthogonal regularization on the temporal dimension "Time_reg" in Equation 8, all three datasets with similarity clustering regularization on the firm dimensions "Sim_reg", and all three datasets with both orthogonal and similarity clustering regularization "Both_reg". Figure 5 signifies the importance of using auxiliary information from multiple datasets and applying regularization on earning prediction. The ML-based prediction with only the sparse and noisy EPS forecast data, even enhanced by advanced data imputation method and sophisticated predictive models cannot beat the simple mean prediction. Combining firm characteristics and return information improves prediction performance, but it is still inferior to the mean. Once we introduce regularization on the time and firm dimensions to reconcile the data heterogeneity and discrepancy, the performance improvement becomes significant. Notably, the similarity clustering regularization on the firm dimension results in the highest (30%) reduction in forecasting error. Compared to those models without regularization, the orthogonal regularization on the time dimension reduces the prediction error by 9%, and the regularizations on both firm and time dimensions further reduce the prediction error by 43%. Figure 5 also illustrates the effects of different ranks on the earning prediction of different versions of NLTCC. With a small rank (= 5), all models' performance is low. Nearly all versions reach the peak performance with a rank between 15 and 20. Once the performance peaks, the marginal benefit of further increasing rank is very low.

Figure 6a presents the running time of two benchmark machine learning models and NLTCC for different tensor ranks. The use of CNN allows NLTCC to avoid heavy operation steps, such as the Kronecker product or Gram matrix [22]. As a result, NLTCC does not incur steep computation/memory costs in the training process. As reflected in the the almost similar computation time between MF and NLTCC. It also shows that regularization reduces

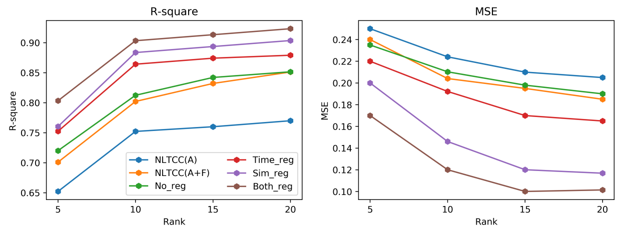
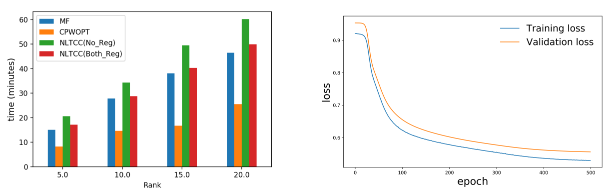


Figure 5: R^2 (left panel) and MSE (right panel) for different models at different ranks.



(a) Running time comparison **(b) Convergence plot**

Figure 6: (a) Running time of different tensor completion algorithms at different ranks (b) Convergence plot of NLTCC.

the computation time for NLTCC as it facilitates a rapid learning process. With a Linux system with 16GB GPU memory, 3854 CUDA cores, and 40 CPU cores, it took NLTCC almost 50 minutes to converge for rank 15. Figure 6b shows the convergence plot in training the NLTCC model. With a learning rate of 0.0001, the model converges in about 400 epochs. Figure 6b shows the convergence plot in training the NLTCC model.

7 CONCLUSIONS

This paper presents NLTCC, a convolutional neural network-based nonlinear tensor coupling and completion framework. NLTCC integrates firm-level characteristics, market return data, and analysts' earnings forecasts to overcome the data quality problems of noise, sparsity, and heterogeneity. We first apply NLTCC to impute missing values in individual analysts' forecasts and then predict the firm's next quarter's earnings with popular machine-learning algorithms. The accuracy of NLTCC-based earnings prediction is significantly higher than the analysts' consensus forecasts. The majority of prediction improvement comes from the superior performance of NLTCC for data pre-processing, data integration, noise removal, and data imputations through reconciling discrepancies among heterogeneous datasets. Extensive experiments on industry sectors and subsectors confirm the effectiveness of the NLTCC model. In particular, NLTCC works exceptionally well in industry sectors with high variance in analysts' forecasts, indicating that a volatile market requires complex information for forecasts and predictions. The backtesting shows that the long-short portfolio based on NLTCC prediction generates much higher returns than the S&P500 index.

REFERENCES

[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106, 1 (2011), 41–56.

[2] Evrim Acar, Tamara G Kolda, and Daniel M Dunlavy. 2011. All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422* (2011).

[3] Faisal M Almutairi, Nicholas D Sidiropoulos, and George Karypis. 2017. Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization. *IEEE Journal of Selected Topics in Signal Processing* 11, 5 (2017), 729–741.

[4] Douglas Ayres, Xuerong Sharon Huang, and Mark Myring. 2017. Fair value accounting and analyst forecast accuracy. *Advances in accounting* 37 (2017), 58–70.

[5] Ryan T Ball and Eric Ghysels. 2018. Automated Earnings Forecasts: Beat Analysts or Combine and Conquer? *Management Science* 64, 10 (2018), 4936–4952.

[6] Daniel Bradley, Sinan Gokkaya, and Xi Liu. 2017. Before an analyst becomes an analyst: Does industry experience matter? *The Journal of Finance* 72, 2 (2017), 751–792.

[7] Francois Brochet, Gregory S Miller, and Suraj Srinivasan. 2013. Do analysts follow managers who switch companies? An analysis of relationships in the capital markets. *The Accounting Review* 89, 2 (2013), 451–482.

[8] Lawrence D Brown. 1993. Earnings forecasting research: its implications for capital markets research. *International journal of forecasting* 9, 3 (1993), 295–320.

[9] J Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* 35, 3 (1970), 283–319.

[10] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[11] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

[12] Lauren Cohen, Andrea Frazzini, and Christopher Malloy. 2010. Sell-side school ties. *The Journal of Finance* 65, 4 (2010), 1409–1437.

[13] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in neural information processing systems*. 155–161.

[14] Eugene F Fama and Kenneth R French. 2006. Profitability, investment and average returns. *Journal of financial economics* 82, 3 (2006), 491–518.

[15] Richard DF Harris and Pengguo Wang. 2019. Model-based earnings forecasts vs. financial analysts' earnings forecasts. *The British Accounting Review* 51, 4 (2019), 424–437.

[16] Richard A Harshman et al. 1970. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. (1970).

[17] Kewei Hou, Mathijs A Van Dijk, and Yinglei Zhang. 2012. The implied cost of capital: A new approach. *Journal of Accounting and Economics* 53, 3 (2012), 504–526.

[18] Shupei Huang, Haizhong An, Xiangyun Gao, and Xuan Huang. 2016. Time–frequency featured co-movement between the stock and prices of crude oil and gold. *Physica A: Statistical Mechanics and its Applications* 444 (2016), 985–995.

[19] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.

[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[21] Charles MC Lee and Eric C So. 2017. Uncovering expected returns: Information in analyst coverage proxies. *Journal of Financial Economics* 124, 2 (2017), 331–348.

[22] Hanpeng Liu, Yaguang Li, Michael Tsang, and Yan Liu. 2019. CoSTCo: A Neural Tensor Completion Model for Sparse Tensors. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 324–334. https://doi.org/10.1145/3292500.3330881

[23] Michael B Mikhail, Beverly R Walther, and Richard H Willis. 1997. Do security analysts improve their performance with experience? *Journal of Accounting Research* 35 (1997), 131–157.

[24] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. 2017. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing* 65, 13 (2017), 3551–3582.

[25] Ledyard R Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 3 (1966), 279–311.

[26] Ajim Uddin, Xinyuan Tao, Chia-Ching Chou, and Dantong Yu. 2020. Nonlinear Tensor Completion Using Domain Knowledge: An Application in Analysts' Earnings Forecast. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 377–384.

[27] Ajim Uddin, Xinyuan Tao, Chia-Ching Chou, and Dantong Yu. 2022. Are missing values important for earnings forecasts? A machine learning perspective. *Quantitative Finance* 22, 6 (2022), 1113–1132.