Domain Adaptive Multi-Modality Neural Attention Network for Financial Forecasting

Dawei Zhou[*]
University of Illinois at
Urbana-Champaign
dzhou21@illinois.edu

Lecheng Zheng
University of Illinois at
Urbana-Champaign
lecheng4@illinois.edu

Yada Zhu
IBM Research
yzhu@us.ibm.com

Jianbo Li
Three Bridges Capital
jianboliru@gmail.com

Jingrui He
University of Illinois at
Urbana-Champaign
jingrui@illinois.edu

ABSTRACT

Financial time series analysis plays a central role in optimizing investment decision and hedging market risks. This is a challenging task as the problems are always accompanied by dual-level (i.e, data-level and task-level) heterogeneity. For instance, in stock price forecasting, a successful portfolio with bounded risks usually consists of a large number of stocks from diverse domains (e.g, utility, information technology, healthcare, etc.), and forecasting stocks in each domain can be treated as one task; within a portfolio, each stock is characterized by temporal data collected from multiple modalities (e.g, finance, weather, and news), which corresponds to the data-level heterogeneity. Furthermore, the finance industry follows highly regulated processes, which require prediction models to be interpretable, and the output results to meet compliance. Therefore, a natural research question is how to build a model that can achieve satisfactory performance on such multi-modality multi-task learning problems, while being able to provide comprehensive explanations for the end users.

To answer this question, in this paper, we propose a generic time series forecasting framework named *Dandelion*, which leverages the consistency of multiple modalities and explores the relatedness of multiple tasks using a deep neural network. In addition, to ensure the interpretability of the framework, we integrate a novel trinity attention mechanism, which allows the end users to investigate the variable importance over three dimensions (i.e, tasks, modality and time). Extensive empirical results demonstrate that *Dandelion* achieves superior performance for financial market prediction across 396 stocks from 4 different domains over the past 15 years. In particular, two interesting case studies show the efficacy of *Dandelion* in terms of its profitability performance, and the interpretability of output results to end users.

CCS CONCEPTS

• **Information systems → Data stream mining**; • **Social and professional topics → Economic impact**.

KEYWORDS

Time Series Forecasting, Heterogeneous Learning, Interpretable Machine Learning

ACM Reference Format:
Dawei Zhou, Lecheng Zheng, Yada Zhu, Jianbo Li, and Jingrui He. 2020. Domain Adaptive Multi-Modality Neural Attention Network for Financial Forecasting. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380288

1 INTRODUCTION

In the financial domain, the aspiration of any investor is to accurately forecast asset value and market behavior with the goal of making the smartest investment decisions. This is a challenging problem as financial systems are usually volatile and influenced by many factors. With the recent big data trend, when traditional data sources, such as quarterly results, earnings calls, and price, being more widely explored, other alternative factors such as satellite imagery and weather provide an edge that investors search for to stay ahead of the pack. For instance, analysts and trading professionals across financial services start leveraging data from satellites to predict Black Friday sales traffic and holiday season results before they are publicly released [8]; one famous study [11] found that relatively cloudier days increase perceived overpricing in individual stocks, and subsequently lead to more selling by institutions.

However, early research on financial forecasting focuses on extracting valuable signals from the alternative data sources (e.g, news [12, 44], social media [18], and Google Trends [39]), while failing to fully leverage the dual-level (i.e, data-level and task-level) heterogeneity. To be specific, the data-level heterogeneity originates from two hierarchies, i.e., the multiple modalities (e.g, finance, weather, and news) on the top and the multiple variables within the same modality at the bottom (e.g, temperature, wind speed, and precipitation within weather). On the other hand, investors usually need to analyze a large number of assets to form a diversified portfolio and mitigate risks. These assets are usually selected from different domains based on industry (e.g, utility, technology and

[*] Part of the work is done as an IBM Research Intern.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380288

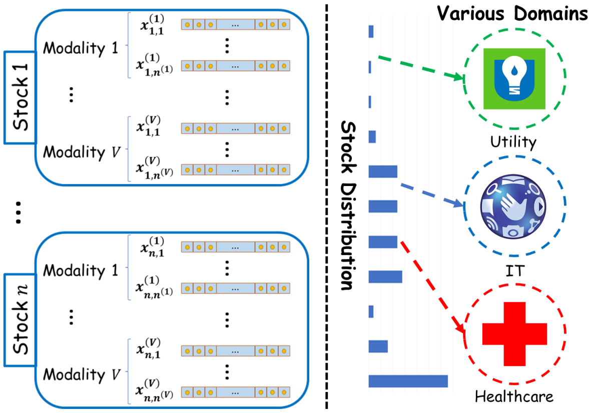


Figure 1: An illustrative example of the multi-modality multi-task time series data from S&P 500 index.

healthcare), and forecasting stocks in each domain can be treated as one task. Thus, this presents the task-level heterogeneity. In Fig. 1, we present an illustrative example of the multi-modalitymulti-task time series data from S&P 500 index. Furthermore, financial services industry requires interpretable models and results to meet compliance and build trust [23]. Despite the tremendous success of deep learning, the finance industry has to rely on traditional decision trees and regression models, that are less effective but much more interpretable to the end users.

Therefore, we have identified the following challenges associated with financial forecasting. First (*C.1 Data Heterogeneity*): how can we model time series that exhibit data heterogeneity? Second (*C.2 Task Heterogeneity*), how can we leverage the potentially noisy input data from various domains to construct models with a satisfactory performance? Third (*C.3 Interpretability*), how do we interpret the output results to the analysts by providing the relevant clues (e.g, the task-specific relevant modalities/variables, the relevant historical time stamps for the future predictions)?

To address these challenges, in this paper, we propose a neural attention network based time series forecasting system named *Dandelion*, which is capable to (i) model multi-modality data, (ii) automatically explore the hierarchical structure regarding task heterogeneity, and (iii) explain the forecasting results to end users. Our proposed *Dandelion* is designed to jointly model the data heterogeneity (*C.1*) and the task heterogeneity (*C.2*) in a principled way. Moreover, to ensure the interpretability of forecasting results to the end users, our *Dandelion* framework integrates a novel trinity attention mechanism (*C.3*) that provides the flexibility for the end users to investigate the importance ratio of the observed data in three dimensions, i.e, tasks, modality variables, time stamps.

The main contributions of this paper are summarized below:

- **Problem.** We formalize the problem of multi-modality multi-task time series forecasting and identify their unique challenges arising from real financial service applications.
- **Model.** We propose a generic neural attention network named *Dandelion*, which is able to jointly leverage the consistency of multi-modality time series and explore the relatedness of multiple tasks. Furthermore, to ensure the interpretability of the output results, we have designed a novel trinity attention mechanism within *Dandelion*, which allows the analysts to investigate the variable importance over three dimensions (i.e, tasks, modality, and time).
- **Evaluations.** Extensive experimental results on 396 real stocks across 14 years demonstrate the performance of the proposed *Dandelion* model. Furthermore, we provide two interesting case studies to show the efficacy of *Dandelion* in terms of the profitability performance and the interpretation of output results to end users in real scenarios.

The rest of our paper is organized as follows. The related work is briefly reviewed in Section 2, followed by the notation and problem definition in Section 3. In Section 4, we formally present our proposed framework *Dandelion*. Experimental results are discussed in Section 5, before we conclude the paper in Section 6.

2 RELATED WORK

In this section, we briefly review the related work regarding multi-view learning, multi-task deep neural networks and interpretable learning in finance.

2.1 Financial Time Series Analysis

Financial time series analysis plays a central role in optimizing investment decision and hedging market risks. Example financial time series analysis includes forecasting stock price, market movement direction and volatility and quarter revenue for a company. Traditionally, statistical methods, such as autoregressive model, moving average model, and their combinations, are widely used. With the development of machine learning techniques in the past decades, artificial neural networks [10, 22], support vector regression [16, 46], deep belief network with restricted Boltzmann machines [26] and LSTM [1] have been applied to forecast future stock prices and price movement direction using historical price data. With the advent of big data era, mining granular signals from alternative data sources, such as news [12, 44] , social media interactions [18], and Google trends [39], to enhance financial time series analysis has gained attention from both the financial industry [8, 47] and the data science community. However, the vast majority (if not all) of existing literature focus on verifying the effectiveness of including *one* particular category of alternative data to analyze financial time series of interest. There lacks generalized methods to simultaneously synchronize the alternative data from different sources to financial time series due to the challenges of data heterogeneity.

2.2 Multi-View Learning

Multi-view learning aims to extract useful information from multiple sources, by exploiting either the consensus principle or the complementary principle to improve the learning performance. Over the decades, many technology and algorithms have been developed to address problems from various domains, such as video surveillance [2], rare category analysis [55], crowd sourcing [57, 58] and social computing [38]. In general, the multi-view learning algorithms can be summarized into three folds: (1) co-training, (2) multiple kernel learning and (3) subspace learning. Notably, co-training [3] is one of the earliest schemes for multi-view learning, which trains alternately to maximize the mutual agreement on two distinct views of the unlabeled data. Co-training based methods have been studied in various contexts, such as active learning [36, 37], label propagation [48], clustering [29], etc. The second category is multiple kernel learning, which exploits kernels that naturally correspond to different views and combine kernels to improve learning performance. For example, in [27], the authors propose a multiple kernel algorithm by solving it as a semi-definite programming problem; in [14], the authors propose a multi-task multi-view learning framework, which can be reduced to standard supervised learning via RKHS. The last category is subspace learning. These types of methods aim to learning a shared latent subspace in order to exploit the relationship among multiple views. [5] presents a multi-view clustering algorithms, by using canoical correlation analysis (CCA) to learn the consensus vectors from different views. Despite the extensive works in muti-view learning, very limited prior art studies the problem of forecasting in multi-view time series data. Recently, [59] proposed a regression model for the multi-view multi-resolution time series data, which estimates the target output by computing the average predictions over all the modalities and all the available resolutions. However, in addition to multiple views, the give observations (i.e, time series) may be collected from various domains that naturally exhibit multiple tasks. To address this issue, in this paper, we propose a domain adaptive multi-view neural attention network *Dandelion* to jointly learn form the view heterogeneity and the task heterogeneity.

2.3 Multi-Task Deep Neural Networks

A surge of research interest on multi-task deep neural networks has been observed in many applications of machine learning, from computer vision [51], nature language processing [34] to person attribute classification [32] and network representation [49]. As summarized in [42], in order to leverage the useful information in multiple related tasks, the existing works are typically done with the advantage of shared hidden layers [4] or shared soft parameters [50]. More recently, in [31], the authors proposed Deep Relationship Network, which placed matrix priors on top of the fully shared hidden layers to learn the relationship between tasks; in [34], the authors introduced a cross-switch unit for the parameter sharing based multi-task deep neural networks, which allowed each task to leverage the relevant knowledge of the other tasks by learning a linear combination of the previous layers. Different from the existing hidden layer sharing based models or soft parameter sharing based models, this paper proposes a domain adaptive neural attention network, which is capable to simultaneously learn from the neural attentions (i.e, soft parameters) and shared hidden layers. Besides, we target the dual heterogeneity (i.e, multiple modalities and multiple tasks) in the application of financial forecasting.

2.4 Interpretable Learning in Finance

Recent years have witnessed the tremendous effort devoted to developing interpretable learning techniques. Existing techniques can generally be categorized into two different groups: designing interpretable models and post-hoc interpretation, depending on the time when the interpretability is obtained [35]. The goal of designing interpretable models is to construct self-explainatory models which incorporate interpretability directly into the structures of a model. In contrast, the post-hoc one requires creating a second model to provide explanations for an existing model. For example, LIME [40] is one of the first work in this direction, which propose a modular and extensible model to faithfully explain the predictions of any model in an interpretable manner; in [24], the authors propose a black-box explanation algorithm, by using influence function to gradually trace the model prediction to the training data and then identify the most responsible example for a given output; Grad-CAM [45] provides a flexible approach for the users to discriminate "strong networks" from the "weak networks"; in [30], the authors developed a generic graph-attention mechanism to provide interpretable inference over the time-evolving graphs. Although many approaches has been proposed, vast majority of existing ones focus on static data, classifiers and applications in computer vision and natural language processing domains. Financial service industry is a heavily regulated industry, where interpretable learning is a vital concern both internally where trust must be built to use increasingly sophisticated models and externally where decisions must meet compliance. Financial service also presents unique challenges, such as data heterogeneity, malicious fraud, temporal dynamics, and user preferences to developing interpretable learning techniques [53, 56, 59]. Up until now, this challenging application area has not attracted much attention from the data mining community. As far as we known, the limited work CLEAR-Trade [25] provides visual interpretations of binary stock market prediction based on attention from the last layer of the deep convolution network. In this paper, we present a trinity attention mechanism that provides the flexibility to end-users to investigate the variable importance over three dimensions (i.e, tasks, modality and time) to enhance the interpretation of prediction results.

3 PROBLEM DEFINITION

Table 1: List of Symbols

Symbol	Definition and Description
$\mathcal{X} = \{\boldsymbol{X_1}, \ldots, \boldsymbol{X_n}\}$	input time series database
$Y = \{\boldsymbol{y_1}, \ldots, \boldsymbol{y_n}\}$	target signal
$\hat{Y} = \{\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_n\}$	prediction of target signal Y
n	number observations (i.e, time series)
v	number of modalityies
$n^{(v)}$	number of variables within the v^{th} modality
T	previous period of relevant time
T'	future time stamps for prediction
$G(\cdot)$	time series forecasting model
$B(\cdot)$	time series forecasting baseline model

Table 1 summarizes the main symbols used in this paper. Throughout this paper, we use lowercase letters to denote scalars (e.g, α), boldface lowercase letters to denote vectors (e.g, $\boldsymbol{v}$), and boldface uppercase letters to denote matrices (e.g, $\boldsymbol{M}$). Suppose we are given a time series data set $\mathcal{X} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_n}\}$ that consists of n observations (e.g, stocks). For each observation $\boldsymbol{X_i}$, $i = 1, 2, \ldots, n$, the data is collected from m modalities, i.e., $\boldsymbol{X_i} = \{\boldsymbol{X_i^{(1)}}, \boldsymbol{X_i^{(2)}}, \ldots, \boldsymbol{X_i^{(m)}}\}$; within each modality, $\boldsymbol{X}_i^{(v)}$, $v = 1, \ldots, m$, we have data from $n^{(v)}$ variables $\boldsymbol{X}_i^{(v)} = \{\boldsymbol{x}_{i,1}^{(v)}, \boldsymbol{x}_{i,2}^{(v)}, \ldots, \boldsymbol{x}_{i,n^{(v)}}^{(v)}\}$; and each variable $\boldsymbol{x}_{i,f}^{(v)}$,

$f = 1, \ldots, n^{(v)}$, is a temporal sequence in the previous relevant time T, i.e, $\boldsymbol{x}_{i,f}^{(v)} = \{x_{i,f}^{(v)}(1), \ldots, x_{i,f}^{(v)}(T)\}$.

Moreover, as the set of time series observations are coming from multiple domains, the importance of variables/modalities may vary dramatically across different domains. For example, in Fig. 1, we present an illustrative example of financial forecasting for the Stardard & Poor's 500 index (S&P 500). In particular, the S&P 500 index is based on the largest companies from various domains, e.g, utility, information technology (IT) and healthcare, etc., listed on the NYSE or NASDAQ; for each stock, the data may be available from different modalities, such as finance, weather, news, web, etc.; within each modality (e.g, finance), the different variables correspond to historical quarterly revenue, consensus, stock price, etc. Without loss of generality, we assume each observation $\boldsymbol{X_i}$ for variable f from modality v at time stamp t has an independent effect on the target signal $\hat{y}_{i,f}^{(v)}(t) = G(\boldsymbol{x}_{i,f}^{(v)})$, where $G(\cdot)$ is a time series forecasting model. Our goal is to learn an accurate forecasting model to produce future T' time stamp predictions

$$\hat{y}_i(\tau) = \sum_{t=1}^{T}\sum_{v=1}^{m}\sum_{i=1}^{n^{(v)}} \hat{y}_{i,f}^{(v)}(t)$$

where $\tau = T+1, \ldots, T+T'$. With the above notation, we formally define our problem as follows:

Problem 1. ***Multi-Modality Multi-Task Time Series Forecasting***
Input: *(i) a multi-modality time series databse* $\mathcal{X} = \{\boldsymbol{X_1}, \ldots, \boldsymbol{X_n}\}$ *from time stamp 1 to time stamp T, (ii) the target signal* $Y = \{\boldsymbol{y_1}, \ldots, \boldsymbol{y_n}\}$ *from time stamp 1 to time stamp T.*
Output: *the prediction* $\hat{Y} = \{\boldsymbol{\hat{y}_1}, \ldots, \boldsymbol{\hat{y}_n}\}$ *from time stamp* $T+1$ *to time stamp* $T+T'$.

4 PROPOSED MODEL

In this section, we present our multi-modality multi-task financial forecasting framework *Dandelion*, which jointly captures the correlations of different variables within each modality, and learns the hierarchical domain representations via a deep neural attention network. In particular, we first show the overall learning paradigm of *Dandelion*, and then discuss the details on modeling the multi-modality multi-task time series data and the capability of interpretation for end users. Finally, we discuss the details of the optimization and implementation of our proposed algorithm.

4.1 A Generic Joint Learning Framework

Our central goal is to develop a generic framework that can model the dual-level heterogeneity and achieve superior results with user-friendly model interpretation. To this end, our framework should take the following three aspects into consideration. First (*C.1*), due to the multi-modality nature of the stock data, our proposed framework should be capable of exploiting the relationships between multiple modalities/variables. Second (*C.2*), there is a huge difference regarding the importance of modalities/variables across stocks from different sectors (e.g, IT, utility, healthcare, agriculture). For example, the modality of weather could play pivotal role in agriculture sector, but have little impact on the IT sector. Therefore, in order to achieve accurate forecasting results, our framework should have the capability to capture the high-level domain knowledge of stocks and understand the underlying importance and logic of modalities/variables across different domains. Third (*C.3*), in addition to the forecasting performance, we also aim to explain the outputs from the proposed model to the end users regarding the identified relevant domains, modalities/variables, and time stamps.

Fig. 2 provides an overview of the proposed *Dandelion* framework. In the Round 1, we develop a neural attention network that is adaptable to model multi-modality time series data, and able to learn stock-specific attention vectors with respect to the relevant time stamps and modalities/variables. Then, we propose an automatic branching procedure for capturing the hierarchical structures of the stock domains. In particular, our *Dandelion* framework gradually makes grouping decisions at each layer from down to top, regarding with whom each task should share the neural attention vectors. This approach is significantly beneficial to hierarchically interpret such bi-level heterogeneous data (i.e, multi-modality, multi-task). In other words, the model is able to not only interpret at the level of modalities/variables (e.g, which variable is more important for healthcare stocks) but also illustrate the correlation and the hierarchical structure of various domains (e.g, the cluster in the granularity of stocks and sectors). Next, we dive into details regarding how *Dandelion* works.

Multivariate forecasting neural network with soft attention. Given a set of multivariate time series $\mathcal{X} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_n}\}$ observed in the previous T time stamps, we propose to use a neural network structure with soft attention [7, 41] for $G(\cdot)$ with respect to the variable f of the i^{th} observation $\boldsymbol{X_i}$ as follows

$$\hat{y}_{i,f}^{(v)}(\tau) = \beta_{i,f}^{(v)}(\tau) h_{i,f}^{(v)}(\tau)$$

where $\beta_{i,f}^{(v)}(\tau)$ and $h_{i,f}^{(v)}(\tau)$ denote the corresponding attention vector and a single hidden layer of dimension d.

$$h_{i,f}^{(v)}(\tau) = \tanh(w_h x_{i,f}^{(v)}(\tau) + b_h)$$

$$a_{i,f}^{(v)}(\tau) = \tanh(w_a h_{i,f}^{(v)}(\tau) + b_a)$$

$$\beta_{i,f}^{(v)}(\tau) = \frac{a_{i,f}^{(v)}(\tau)}{\sum_{i=1}^{n}\sum_{v=1}^{m}\sum_{f=1}^{n^{(v)}}\sum_{t=1}^{T} a_{i,f}^{(v)}(t)}$$

where w_h and b_h refer to the weight and bias in the hidden layer; w_a and b_a represent the weight and bias in the attention layer. Note that the above single-layer structure can be naturally generalized to deeper neural networks.

Intuitively, the attention vector $\boldsymbol{\beta_{i,f}^{(v)}}$ can be interpreted as the summary impact [7] on the variable f of the v^{th} modality i^{th} observation $\boldsymbol{x_{i,f}^{(v)}}$ in the context of other observations. As each observation is considered to have an independent impact on the forecast prediction, the soft attention mechanism aims to generate the common logic a human would follow. To be more specific, the attention mechanism first computes $\boldsymbol{a_{i,f}^{(v)}}$ that measures how important the observation is; then, by incorporating the individual impact of each observation, the neural attention network extracts a small subset of observations that are most important and could be decisive to the prediction. We have also considered a series of other neural

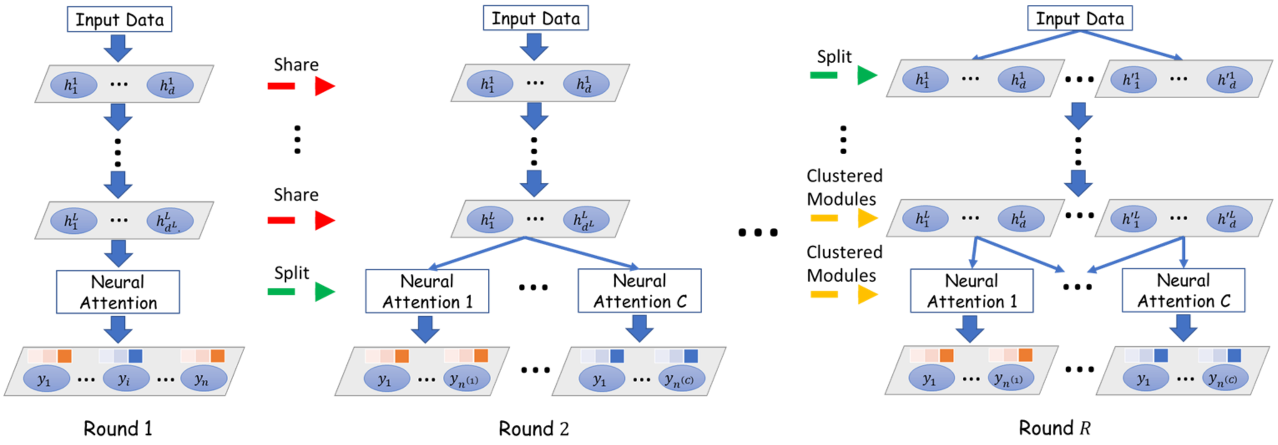


Figure 2: An illustration of the proposed *Dandelion* framework. In Round 1, we construct a neural attention network with L layers and derive the affinity matrix for the L^{th} layer by computing the cosine similarities over each pair of attention vectors. At the beginning of Round 2, we first group the n stocks into c^L clusters based on the affinity matrix. Then, we clone the $(L-1)^{\text{th}}$ layer by directly copying the hidden weights of neural network for $c^L - 1$ time, and link the n observations to c^L neural networks in the $(L-1)^{\text{th}}$ layer. In the following Rounds $l = 3, \ldots, R$, we keep compute the affinity matrix for grouping the hidden units in the previous $(l+1)^{\text{th}}$ layer, and then perform the split and widen procedure in the current l^{th} layer. We repeat this procedure until the layer could not be further divided or it has reached the top of the neural network.

network mechanisms, such as recurrent neural network, to serve as the base model. However, based on our empirical studies, there is little gain in performance, but a large increase in the model and computational complexity. In the highly regulated industries like finance, we believe such increased computational efficiency is well worth especially for the extensive end user analysis. In the following section, we further design the cost function of the above neural attention network to addressing the $(C.1)$ data heterogeneity.

Learning from multi-modality time series data. As illustrated in Fig. 1, the input time series naturally forms a 2-level hierarchy, where the multiple modalities are on the top, and the multiple variables are at the bottom. Without loss of generality, we make the following basic assumptions: (1) the information from different modalities is complementary [28], whereas the variables within the same modality are redundant in forecasting the output signal; (2) for a given observation $\boldsymbol{X_i}$, only a relatively small subset of variables are relevant to making the prediction at a certain time stamp. For example, in predicting the actual price of IT companies (e.g, Apple inc.), different modalities (e.g, finance, weather, news, etc.) provide redundant information for the output in different aspects; among all the observed variables, the finance and news variables are the most relevant ones. Based on these assumptions, we propose to enforce the consensus of variables within the same modality and the sparse attention over modalities/variables for the sake of model robustness. In particular, our multi-modality neural attention network is formulated as follows.

$$\mathcal{L}(\tau) = \mathcal{L}_Y(\tau) + \mathcal{L}_s(\tau) + \mathcal{L}_c(\tau) \tag{1}$$

$$= \underbrace{\sum_{i=1}^{n} |y_i(\tau) - \hat{y}_i(\tau)|}_{\mathcal{L}_Y\text{: prediction loss}} \underbrace{+\gamma \sum_{i=1}^{n} \sum_{v=1}^{m} \sum_{f=1}^{n^{(v)}} \sum_{t=1}^{T} |a_{i,f}^{(v)}(t)|}_{\mathcal{L}_s\text{: sparse attention regularizer}}$$

$$\underbrace{+\eta \sum_{i=1}^{n} \sum_{v=1}^{m} \sum_{f=1}^{n^{(v)}} |z_i^{(v)}(\tau) - \beta_{i,f}^{(v)}(\tau)|}_{\mathcal{L}_c\text{: consensus regularizer}}$$

where $\boldsymbol{z_i^{(v)}}$ is the consensus embedding, and γ, η are hyper parameters to balance the impact of this term on the overall objective function. To be more specific, for timestamp τ, the first term $\mathcal{L}_Y$ measures the prediction error via mean squared error; the second term $\mathcal{L}_s$ corresponds to the sparse regularizer, where an L_1 norm is adopted over the unnormalized attention vectors $\boldsymbol{a_{i,f}^{(v)}}$ to select key modalities/variables for each observation in the previous T timestamps; the third term is the consensus regularizer, which enforces the consistency across variables within the same modality by mapping all the normalized attention vectors $\boldsymbol{\beta_{i,f}^{(v)}}$ to a consensus embedding $\boldsymbol{z_i^{(v)}}$ for each modality v. To address $(C.2)$ task heterogeneity, we introduce a fully-adaptive hierarchical clustering strategy for multi-task learning in the following section.

Fully-adaptive hierarchical multi-task learning. Intuitively, different stocks from the same domain may share similar attention vectors and thus exhibit similar patterns. For example, most healthcare stocks rely on the news from Food and Drug Administration.

This is because a positive report regarding a stock will boost the price of this stock rapidly in a short time, while an accident report will put the stock in jeopardy. Similarly, some stocks in consumer discretionary sector (e.g, Walmart, McDonald etc.) might rely on the variables of the weather. On the other hand, stocks from distinct sectors may also share similar patterns. Similar to some stocks from consumer discretionary sector, weather might also play an important role in financial forecasting for some stocks from energy sector, such as Cabot Oil & Gas, because the freezing weather during the winter results in the increase the demand of the natural gas and then might boost the price of these stocks.

Based on this observation, we exploit the relatedness of attention vectors for different stocks or tasks by grouping similar stocks/tasks into the same cluster. Our training algorithm involves a procedure to split and widen the layers of neural networks, as used in [32, 54]. For the sake of explanation, we call the layer that we want to split and widen as the *split layer*. Suppose that we have L layers in neural network in total, and we start the split and widen procedure from layer L up to layer 1 (shown in Fig. 2). At first, we compute and record the attention vector for each individual observation $\boldsymbol{X}_i, i = 1, \dots, n$. Then, we derive the affinity matrix $\boldsymbol{A}^L \in R^{n \times n}$ at layer L by computing the cosine similarities of each pair of the weights of the neural attention network. We calculate the similarity between the weight of the i^{th} branch $\boldsymbol{w}_i$ and the weight of the j^{th} branch $\boldsymbol{w}_j$ as

$$\boldsymbol{A}(i,j)^l = \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \, \|\boldsymbol{w}_j\|} \tag{2}$$

where $l = 1, \dots, L$, $\|\cdot\|$ represents the L_2 norm.

After obtaining the similarity matrix, we determine the optimal number of clusters we assign these tasks to, by minimizing the loss function as follows:

$$\mathcal{L}_c^l = \mathcal{L}_{sc} + c^l(\alpha)^l \tag{3}$$

where $\alpha \in (0, 1)$ is a positive parameter and $\mathcal{L}_{sc}$ is the total spectral clustering loss [32]. The second part of this equation is a penalty term, which constrains the number of clusters c^l at the l^{th} split layer. Intuitively, the loss increases as we update the network structure from the L^{th} layer to the 1st layer, thus the number of clusters is decreasing from the down to the top in Fig. 2. At the beginning of Round 2 in Fig. 2, the L^{th} layer becomes the split layer. We decompose this unit layer into c^L branches and back link to the $(L-1)^{\text{th}}$ layer. In general, the weight of each newly-created branch at layer l is initialized by directly copying the weight from the current split layer. In other words, we clone $(c-1)^l$ branches at layer l and re-link the neural network. Next, we re-train the updated neural network for some iterations and follow the same procedure to find the similarities of each pair of branches by computing the cosine similarities of their weights. We repeat this procedure until the branches cannot be divided or the split layer reaches the top layer of the neural network.

Algorithm 1 Domain Adaptive Multi-Modality Attention Network (*Dandelion*)

```
Input:
    (1) Multi-modality time series $\mathcal{X} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_n}\}$;
    (2) History data of the target signal $Y = \{\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_n}\}$;
    (3) Forecasting baseline model $B(\cdot)$;
    (4) Previous relevant time $T$;
    (5) Total number of training round $R$.
Output:
    Predictions of target signal $\hat{Y} = \{\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \dots, \hat{\boldsymbol{y}}_n\}$ in the future $T'$ time stamps.
Pre-train the proposed neural attention network and compute the independent attention vector $\boldsymbol{\beta_i}$ for each observation $\boldsymbol{X_i}$.
Let $l = L$, $r = 1$.
while Stopping criterion is not satisfied and $r \le R$ do
    Compute the affinity matrix $\boldsymbol{A^l}$ for $l^{\text{th}}$ layer in Eq. 2.
    Determine the number of clusters by minimize $\mathcal{L}_c^l$ in Eq. 3.
    Create branches in the $l^{\text{th}}$ layer and update the network structures in the $l^{\text{th}}$ and $(l-1)^{\text{th}}$ layers.
    Update the hidden layers' parameters by minimizing $\mathcal{L}(\tau)$ in Eq. 1.
    Let $l \longleftarrow l - 1$, $r \longleftarrow r + 1$.
end while
```

End-user oriented interpretation via trinity attention. In learning a forecasting model over multi-modality multi-task time series data, the interpretability of the predictive model is critical for end users to understand and evaluate the model outputs. However, many existing time series forecasting models (e.g, autoregressive integrated moving average [52], long short-term memory [15], the gated recurrent unit [6], etc.) fall short of the interpretability for multi-modality multi-task time series data (illustrated in Fig. 1).

To address this issue, we develop a comprehensive trinity attention mechanism, which learns the independent importance weight over tasks (e.g, stock sectors), time and modalities/variables. In particular, based on the aforementioned fully-adaptive multi-task learning mechanism, the stocks automatically form R-level hierarchical clusters; each cluster $C^{(l)}$ at the l^{th}-level represents a set of stocks that share the same or similar attention matrix $\boldsymbol{\beta} \in \mathbb{R}^{T \times F}$, where $F = \sum_{v=1}^{m} n^{(v)}$ denotes the total number of variables, and each entry in $\boldsymbol{\beta}$ indicates the variable-wise temporal importance for predicting the target signals $\boldsymbol{Y}$. Moreover, in order to extract distinguishable attention distribution for end users, we adopt the summarization function proposed in [13] $f_{agg} : \mathbb{R}^{A \times B} \to \mathbb{R}^{B}$ to independently quantify the temporal importance distribution and the variable importance distribution as follows

$$\boldsymbol{\beta}_{var} = f_{agg}(\boldsymbol{\beta}) = \left[\frac{\sum_t^T \beta(t,1)}{\sum_t^T \sum_f^F \beta(t,f)}, \dots, \frac{\sum_t^T \beta(t,F)}{\sum_t^T \sum_f^F \beta(t,f)} \right] \tag{4}$$

$$\boldsymbol{\beta}_{temp} = f_{agg}(\boldsymbol{\beta}^T) = \left[\frac{\sum_f^F \beta(1,f)}{\sum_t^T \sum_f^F \beta(t,f)}, \dots, \frac{\sum_f^F \beta(T,f)}{\sum_t^T \sum_f^F \beta(t,f)} \right] \tag{5}$$

where the unified variable-wised attention vector follows $\sum_f^F \beta_{var}(f) = 1$, $\beta_{var}(f) \in [0,1]$; and the unified temporal-wise attention vector follows $\sum_t^T \beta_{temp}(t) = 1$, $\beta_{temp}(t) \in [0,1]$.

4.2 Optimization Algorithm

In our implementation, our proposed *Dandelion* framework is trained with Stochastic Gradient Descent (SGD) until convergence

Sector	# of stocks	Starting time stamp	Endding time stamp
Consumer Cyclical	90	5-6-2004	6-26-2018
Healthcare	105	5-3-2004	5-20-2018
Industrial	98	5-4-2004	6-27-2018
Technology	103	5-3-2004	6-25-2018

Table 2: Statistics of the stock data sets.

on the validation set. The optimization algorithm is summarized in Algorithm 1. The inputs of Algorithm 1 include the observed time series $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$, the history data of the target signal $Y = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$, the forecasting baseline model $B(\cdot)$, the number of relevant historical time stamps T and the number of training round R. The algorithm works as follows. We first pretrain our proposed neural attention mechanism (Eq. 1) and compute the attention vector $\boldsymbol{\beta}_i$ for each time series observations $\boldsymbol{X}_i$ independently. Then, Step 3 to Step 9 is the main body of the learning process, which gradually explores the hierarchical domain knowledge over the observations. In particular, at each layer l, we compute the affinity matrix $\boldsymbol{A}^l$ regarding the task similarities. By determining the number of clusters (i.e, branches) based on $\mathcal{L}_c^l$ in Eq. 3, we generate branches and assign similar tasks into the same branch. The algorithm stops when the stopping criterion (e.g, maximum running time, error rate lower bound) is satisfied and $r > R$.

5 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed model *Dandelion* in terms of prediction accuracy, profitability and interpretability for end users.

5.1 Experiment Setup

Data sets: We collect finance data, news, Google Trends[1] and weather data for 396 public US companies to evaluate the *Dandelion* framework. The data origins from 90 companies from consumer cyclical, 105 companies from healthcare, 98 companies from industrial, and 103 companies from technology. The data ranges from May 3, 2004 to June 27, 2018. Note that some companies have less data due to short history.

The finance data includes historical quarterly revenue, consensus, stock price and various of their derivatives. Revenue growth is the key indicator of the valuation and profitability of a company and a major input for long-term value-based investment strategy. Revenue growth measures a company's earning power and stock performance. Large brokerage firms employ legions of stock analysts to publish forecast reports on companies' earnings (revenue minus expense) over the coming years. A few companies (e.g. Thomson Reuters) compile the estimates and compute the average or median as *consensus*. The consensus number can be adjusted at any time point before the actual revenue is announced. The historical revenue and consensus estimate data is obtained from SEC[2] and Yahoo! Finance, respectively. Due to the long tail distribution of revenue growth, we set the target time series as revenue growth minus consensus which is acknowledged as *revenue surprise* in the investment communities. Note revenue surprise can be either positive or negative. The daily historical and forecasted weather data is collected from The Weather Company[3]. The data includes maximum, minimum, average values of temperature, wind, pressure, precipitation and cloud. The weather data is aggregated when a company has multiple locations using spatio statistics. The news data includes the data used in [9] and The New York Times[4]. We link each news article to a company as long as the company is mentioned. We create a set of features from each news article using sentiment analysis, taxonomies, and document summarization techniques. The Google Trend is extracted based on daily total number of hits on company names, major products and executives. The detailed feature engineering process is out the scope of this paper.

In summary, we simultaneously forecast the quarterly revenue surprise (zero or near-zero mean) of the 396 companies at the daily level before their revenue is announced using the up-to-date information derived from finance, news, Google Trends and weather. Despite revenue is published each quarter, daily forecast of revenue surprise enable investors to adjust their portfolio granular for return and risk analysis.

Comparison methods: We compare the proposed *Dandelion* framework with following benchmark methods.

- **[ConEst:]** the Wall Street consensus estimates as discussed in the previous part of this subsection.
- **[ARIMAX [17]:]** an Auto Regressive Integrated Moving Average based method, which is a special case of vector auto-regression in the context of time series forecasting.
- **[MVR [20]:]** a multi-view regression approach that uses canonical-correlation analysis (CCA) to learn the consensus representation and makes predictions via ridge regression.
- **[Bi-LSTM [33]:]** a bi-directional LSTM architecture, where the input time series is fed in normal time order for one network, and in reverse time order for another.
- **[MNA [41]:]** a neural attention network that is designed for demand forecasting using multi-modality event data.
- **[*Dandelion*-M:]** a variation of our *Dandelion* framework, which ignores the task heterogeneity and simply optimize model using the objective function in Eq. 1.
- **[*Dandelion*-D:]** a variation of our *Dandelion* framework, which ignores the data heterogeneity but adopts the hierarchical multi-task learning mechanism. Compared with *Dandelion*, the multi-modality consensus regularizer $\mathcal{L}_c$ is not included in the objective function.

Repeatability: In our experiments, we let the hyper parameters $\alpha = 0.5, \gamma = 1, \eta = 1$. The experiments are performed on a Windows machine with four 3.5GHz Intel Cores and 256GB RAM.

5.2 Prediction Performance

We compare the prediction accuracy based on the median absolute deviation [43] (Med-abs) which is robust to outliers. We split the data into train (80%) and test (20%) sequentially. Table 3 summarizes the prediction performance on the test set across all the companies

[1] https://trends.google.com/trends/

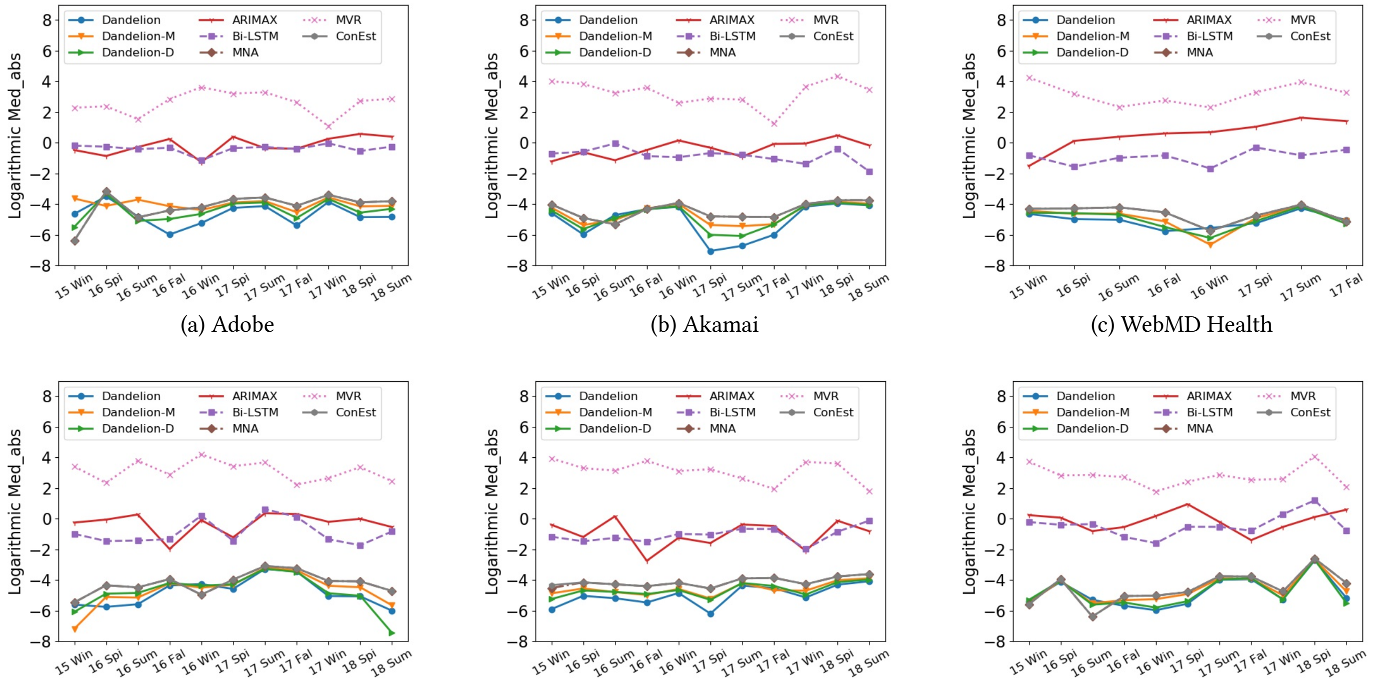
[2] https://www.sec.gov

[3] https://weather.com/

[4] https://www.nytimes.com/

Methods		Con. Cyc.	Healthcare	Indus.	Tech.	All
Industry Benchmark	ConEst	0.01575	0.02247	0.01587	0.02133	0.01857
Regression	ARIMAX	1.22291	1.95461	1.28935	2.08068	1.55457
	MVR	0.48691	0.48922	0.51235	0.57606	0.51599
Neural Networks	Bi-LSTM	0.93098	1.54184	0.97901	1.44376	1.19222
	MNA	0.01692	0.02251	0.01695	0.02132	0.01960
Our Approaches (v.s ConEst)	*Dandelion*	0.01430 (↓ 9.2%)	**0.02119** (↓ 5.7%)	**0.01560** (↓ 1.7%)	**0.01883** (↓ 11.7%)	**0.01731** (↓ 6.8%)
	Dandelion-M	0.01582 (↑ 0.4%)	0.02173 (↓ 3.2%)	0.01579 (↓ 0.5%)	0.02032 (↓ 4.8%)	0.01806 (↓ 2.8%)
	Dandelion-D	**0.01387** (↓ 11.9%)	0.02127 (↓ 5.3%)	0.01567 (↓ 1.3%)	0.01970 (↓ 7.6%)	0.01753 (↓ 5.6%)

Table 3: Results of four sector companies. *Dandelion* and its variations (i.e, *Dandelion*-M, *Dandelion*-D) achieve smaller Med-abs values than all benchmark methods on each individual sector as well as the overall performance. (The lower the better)



(a) Adobe

(b) Akamai

(c) WebMD Health

(d) Red Hat

(e) Salesforce

(f) PVH

Figure 3: Individual prediction performance of six companies over time. Dandelion consistently performs better than all other methods in most of the time. (The lower the better)

as well as that by sectors (i.e, consumer cyclical, healthcare, industrial, technology). From Table 3, we observe that (1) *Dandelion* and its variations (i.e, *Dandelion*-M and *Dandelion*-D) reduce the Med-abs errors of ConEst up to 11.9% across the four sectors of stocks. (2) *Dandelion* and its variations (i.e, *Dandelion*-M and *Dandelion*-D) achieve smaller Med-abs errors than all the benchmark methods on each individual sector as well as the overall performance including companies all the sectors. In particular, for the Technology sector, *Dandelion* obtains 99% smaller Med-abs than ARIMAX, 97% smaller than MVR, 98% smaller than Bi-LSTM, and 15% smaller than MNA. Besides, we also perform a t-test of the Med-abs results between the *Dandelion* and our best competitor ConEst on all the stocks across all the 396 stocks. The p−value is $3.015e{-}19$, which suggest the improvement of *Dandelion* is statistically significant; (3) *Dandelion* outperforms its two variations that either only models data heterogeneity (*Dandelion*-M) or task heterogeneity (*Dandelion*-D) in three out of four sectors. The superior performance of *Dandelion* validates the effectiveness of jointly leveraging the consistency of multi-modality time series and exploring the relatedness of multuple tasks. *Dandelion* achieves slightly higher Med-abs than *Dandelion*-D, but still smaller than other methods, in the consumer cyclical sector. This might be due to the large variations of companies within this sector which might introduce noise to the multi-modality consensus regularizer. To further demonstrate the

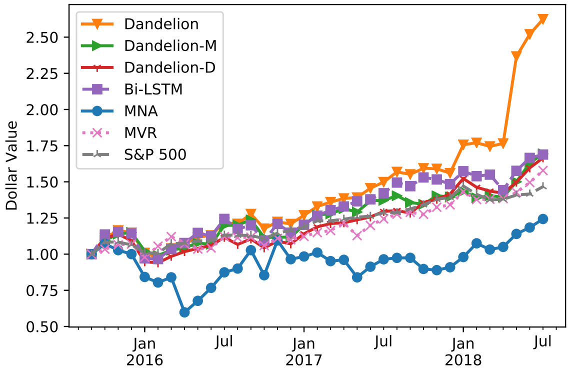


Figure 4: Portfolio value across the testing period if starting with $1. *Dandelion* outperforms all the benchmark portfolios and increased more than 1.6 times in less than 3 years. (The larger the better)

performance of *Dandelion*, in Fig. 3, we present the individual forecasting results of selected companies (i.e, Adobe, Akamai, WebMD Health, Red Hat, Salesforce, PVH) by quarter in the test period. We can see that *Dandelion* consistently performs better than all other methods in most of the time.

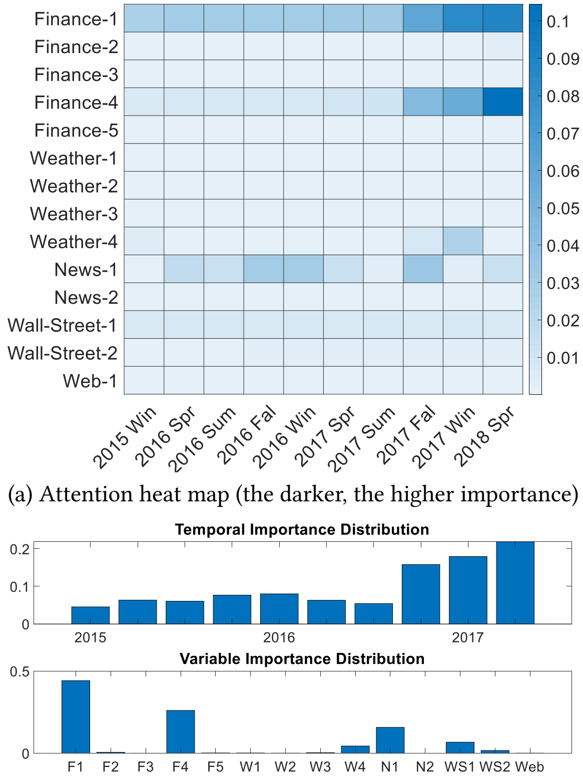
5.3 Profitability Performance

Taking the prediction results on revenue surprise in the previous experiments, we evaluate the profitability performance based on a simulated portfolio. Note for the sake of illustration, we choose a simple way to construct the portfolio. To be specific, we select the top $K(0 < K \leq 396)$ companies with the highest and positive revenue surprise forecast at the beginning of the test period. We set the weight of each asset proportionally to the revenue surprise forecast as revenue surprise is positively correlated to future stock price movement [19, 21]. Suppose the portfolio starts with a unitary value and the weight of each asset is updated at the end of each month during the test period based on the latest revenue surprise forecast. We plot the portfolio value of all the methods but ARIMAX over the test periods for $K = 20$ in Fig. 4. ARIMAX yields a portfolio concentrates on very few assets and much more volatile than the others which skews the figure's scale. Fig. 4 shows that *Dandelion* not only beats *S&P* 500 index but also yields the most total return which is 1.6 times more than the original value during the test period. For other K values, we observe the similar pattern but skip the presentation due to space limitation. This validates the superior performance of *Dandelion* in generating investment returns.

Taking one step further, we calculate the Sharpe ratio of the simulated portfolios. Sharpe ratio measures investment performance adjusted by risk which is one of the most important characteristics of a portfolio. Sharpe ratio usually falls within the range of 0 to 3, the larger the better. As shown in Tab. 4, *Dandelion* obtains the highest Sharpe ratio values with respect to different K values. During the same period, the Sharpe ratio of *S&P* 500 Index is 1.49. Thus for most cases, *Dandelion* beats *S&P* 500 Index, but not any other methods do. Following *Dandelion*, *Dandelion*-M and MVR get the second and third largest Sharpe ratios given all K values. Both approaches explicitly incorporate data heterogeneity. *Dandelion*-D: adopts the hierarchical multi-task learning mechanism and obtains the fourth largest Sharpe ratio. These results not only demonstrate the superior performance of the *Dandelion* framework but also the value of simultaneously accounting for the dual-heterogeneity within the *Dandelion* framework.

Note that the portfolio construction adopted here is for demonstration purpose and hence relatively simple and should be further optimized before utilized in real life trading. One example is missing of transaction cost. Another example is that the score weighted system could lead to concentrated portfolios. Actually, the big jump observed in Fig. 4 around may 2018 in *Dandelion*is an example of over-concentrated positions. However, even after remove these obvious jumps, it is pretty straightforward to see that *Dandelion* outperforms other methods.

5.4 Interpretation for End Users



(a) Attention heat map (the darker, the higher importance)

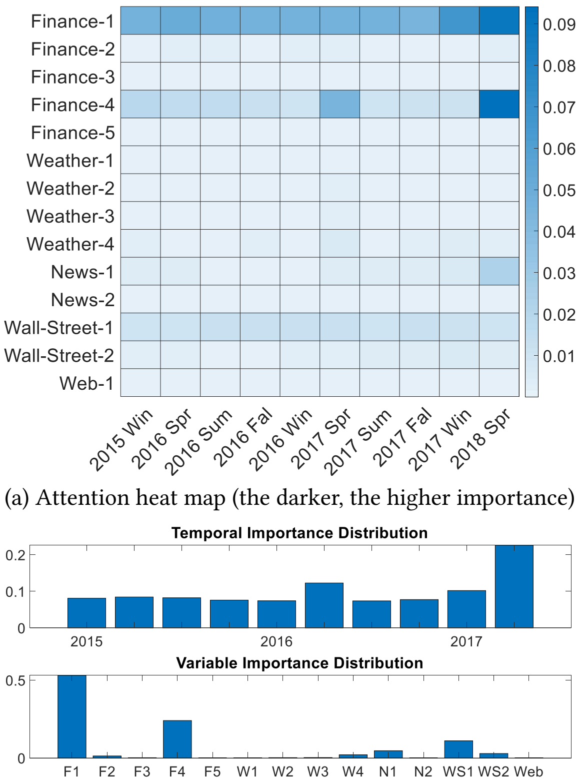
(b) Summary attention

Figure 5: Interpretation of Amgen.

Dandelion integrates a novel trinity attention mechanism, which allows the end users to interpret prediction outputs with respect to variable importance over three dimensions (i.e., tasks, modality and time). To the best of our knowledge, there is little work unveil such complex and fine-grained interpretation in the context of multi-modality multi-task time series data. In this subsection, we demonstrate interpretation capability provided by *Dandelion* using

Portfolio Size	10	20	30	40	50	60	All Positive
Dandelion	**1.57**	**1.49**	**1.43**	**1.50**	**1.45**	**1.51**	**1.61**
Dandelion-M	1.23	1.21	1.31	1.37	1.38	1.44	1.51
Dandelion-D	0.94	1.14	1.13	1.19	1.20	1.25	1.39
ARIMAX	0.90	0.90	0.90	0.90	0.90	0.90	0.90
Bi-LSTM	0.95	1.02	1.04	1.06	1.09	1.10	1.12
MNA	0.33	0.39	0.50	0.55	0.58	0.62	0.85
MVR	1.08	1.22	1.24	1.27	1.27	1.26	1.29

Table 4: Sharpe ratios of the simulated portfolios under the strategies derived from the studied prediction methods. *Dandelion* yields the best portfolios across all different sizes. (The higher the better)



(a) Attention heat map (the darker, the higher importance)

(b) Summary attention

Figure 6: Interpretation of Total System Services.

case studies. For example, Fig. 5(a) shows the attention vectors of Amgen[5], a company in the healthcare sector, during the test period. In this attention map, the x-axis is the time, the y-axis is associated with variables from multiple modalities, and the darker the color, the larger the attention value, i.e. more important. This attention heat map provides end users visualized explanation of the driven variables to the prediction output over time, allowing end uses to target on specific time and variables for further investigation. The attention vectors can be aggregated along time or variable dimensions to provide end users a higher level view via Eq. 4 and Eq. 5, as shown in Fig. 5(b). In the case of Amgen, the summary attention explains the prediction output as follows: (1) the temporal importance is increasing over time, which indicates the relevance of data is higher when the time is closer to the forecasting time stamp; (2) the finance variables play a central role for predicting revenue surprise. Moreover, by comparing the attention heat maps and summary attention of different companies in different domains, it can guide end users to investigate the correlation and difference between multiple tasks. For example, comparing Fig. 5 and Fig. 6, we can see that the news modality is more important to Amgen than to Total System Services. According to Bloomberg[6], during Jan 2015 to Jan 2019, the ratio between the average numbers of tweets and daily news on Amgen is 1.44 (107/74) vs 0.43 (12/28) on Total System Service. This implies that people are more interested in the news related to Amgen than Total System Service and news has more impact to the stock of Amgen than Total System Service. This sheds lights on market analysis.

6 CONCLUSION

We have presented a novel neural attention based framework (*Dandelion*) for financial time series forecasting problem with both data heterogeneity and task heterogeneity. To accommodate the model explanation for end users, we propose the trinity attention mechanism that provides the flexibility for users to investigate the variable importance over three dimensions (i.e., tasks, modality and time). We demonstrate the effectiveness of *Dandelion* in financial market forecasting problem cross 396 real stocks from 4 different domains over the past 15 years. Moreover, our model offers superior profitability performance and descriptive capabilities in comparison to the state-of-the-art methods.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under Grant No. IIS-1552654 and Grant No. IIS-1813464, the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-02-00 and Ordering Agreement Number HSHQDC-16-A-B0001, an IBM Faculty Award, and IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

[5] https://www.amgen.com/

[6] https://www.bloomberg.com/

REFERENCES

[1] W. Bao, J. Yue, and Y. Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE* (2017).

[2] J. Black, T. Ellis, and P. Rosin. 2002. Multi view image surveillance and tracking. In *MVC (2002)*.

[3] A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT (1998)*.

[4] R. Caruana. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *ICML (1993)*.

[5] K. Chaudhuri, S. M Kakade, K. Livescu, and K. Sridharan. 2009. Multi-view clustering via canonical correlation analysis. In *ICML (2009)*.

[6] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[7] J. K Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. 2015. Attention-based models for speech recognition. In *NIPS (2015)*.

[8] B. Cotton. 2019. Is 'Alternative Data' becoming the new normal for investors? Business Leader Interviews.

[9] X. Ding, Y. Zhang, T. Liu, and J. Duan. 2014. Using structured events to predict stock price movement: An empirical investigation.. In *EMNLP (2014)*.

[10] M. Ghiassi, J. Skinner, and D. Zimbra. 2013. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *ESA* (2013).

[11] W. N. Goetzmann, D. Kim, A. Kumar, and Q. Wang. 2015. Weather-Induced Mood, Institutional Investors, and Stock Returns. *The Review of Financial Studies* (2015).

[12] Fung GPC, Yu JX, and Lu H. 2005. The predicting power of textual information on financial markets. *IIB* (2005).

[13] T. Guo and Tao Lin. 2018. Exploring the interpretability of LSTM neural networks over multi-variable data. (2018).

[14] J. He and R. Lawrence. 2011. A Graphbased Framework for Multi-Task Multi-View Learning.. In *ICML (2011)*.

[15] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997).

[16] W. Huang, Y. Nakamori, and S.-Y. Wang. 2005. Forecasting stock market movement direction with support vector machine. *COR* (2005).

[17] R. J Hyndman and G. Athanasopoulos. 2018. *Forecasting: principles and practice.* OTexts.

[18] Bollen J, Mao H, and Zeng X. 2011. Twitter mood predicts the stock market. *JCS* (2011).

[19] N. Jegadeesh and J. Livna. 2006. Revenue surprises and stock returns. *JAE* (2006).

[20] S. M Kakade and D. P Foster. 2007. Multi-view regression via canonical correlation analysis. In *COLT (2007)*.

[21] I. Kama. 2009. On the Market Reaction to Revenue and Earnings Surprises. *JBFA* (2009).

[22] Y. Kara, M. A. Boyacioglu, and Ö. K. Baykan. 2011. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *ESA* (2011).

[23] W. Knight. 2017. The financial workd wants to open AI's black boxes. Intelligent Machines.

[24] P. W. Koh and P. Liang. 2017. Understanding black-box predictions via influence functions. *ICML (2017)* (2017).

[25] D. Kumar, G. W Taylor, and A. Wong. 2017. Opening the Black Box of Financial AI with CLEAR-Trade: A CLass-Enhanced Attentive Response Approach for Explaining and Visualizing Deep Learning-Driven Stock Market Prediction. *arXiv preprint arXiv:1709.01574* (2017).

[26] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashia. 2014. Time series forecasting using a deep belief network with restricted Boltzmann machines. *Neurocomputing* (2014).

[27] G. RG Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I Jordan. 2004. Learning the kernel matrix with semidefinite programming. *JMLR* (2004).

[28] J. Li, J. He, and Y. Zhu. 2017. HiMuV: Hierarchical Framework for Modeling Multi-modality Multi-resolution Data. In *ICDM (2017)*.

[29] S. Li, M. Shao, and Y. Fu. 2015. Multi-view low-rank analysis for outlier detection. In *ICDM (2015)*.

[30] Z. Liu, D. Zhou, and J. He. 2019. Towards Explainable Representation of Time-Evolving Graphs via Spatial-Temporal Graph Attention Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 2137–2140.

[31] M. Long and J. Wang. 2015. Learning multiple tasks with deep relationship networks. *CoRR* (2015).

[32] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. S. Feris. 2017. Fully-Adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification. (2017).

[33] X. Ma and E. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354* (2016).

[34] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. 2016. Cross-stitch networks for multi-task learning. In *CVPR (2016)*.

[35] G. Montavon, W. Samek, and K.-R. Müller. 2018. Methods for interpreting and understanding deep neural networks. *DSP* (2018).

[36] I. Muslea, S. Minton, and C. A Knoblock. 2002. Active+ semi-supervised learning= robust multi-view learning. In *ICML (2002)*.

[37] I. Muslea, S. Minton, and C. A Knoblock. 2003. Active learning with strong and weak views: A case study on wrapper induction. In *IJCAI (2003)*.

[38] S. J. Pan, J. T Kwok, Q. Yang, and J. J. Pan. 2007. Adaptive localization in a dynamic WiFi environment through multi-view learning. In *AAAI (2007)*.

[39] T. Preis, H. S. Moat, and H. E. Stanley. 2013. Quantifying Trading Behavior in Financial Markets Using Google Trends. *Scientific Reports* (2013).

[40] M. Ribeiro, S. Singh, and C. Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *SIGKDD (2016)*.

[41] M. Riemer, A. Vempaty, F. Calmon, F. Heath, R. Hull, and E. Khabiri. 2016. Correcting forecasts with multifactor neural attention. In *ICML (2016)*.

[42] S. Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR* (2017).

[43] D. Ruppert. 2011. *Statistics and data analysis for financial engineering.* Vol. 13. Springer.

[44] R. Schumaker and H. Chen. 2006. Textual analysis of stock market prediction using financial news articles. In *AMCIS*.

[45] R. R Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV (2017)*.

[46] Alaa F. Sheta, Sara Elsir M. Ahmed, and Hossam Faris. 2015. A Comparison between Regression, Artificial Neural Networks and Support Vector Machines for Predicting Stock Market Index. *IJARAI* (2015).

[47] L. Stevens. 2018. Alternative Data: How to Find Signal in the Noise. FACTSET.

[48] W. Wang and Z.-H. Zhou. 2010. A New Analysis of Co-Training.. In *ICML (2010)*.

[49] L. Xu, X. Wei, J. Cao, and P. S. Yu. 2017. Multi-task Network Embedding. In *DSAA (2017)*.

[50] Y. Yang and T. M. Hospedales. 2016. Trace Norm Regularised Deep Multi-Task Learning. *CoRR* (2016).

[51] A. R Zamir, A. Sax, W. Shen, L. J Guibas, J. Malik, and S. Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *CVPR (2018)*.

[52] G P. Zhang. 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* (2003).

[53] S Zhang, D Zhou, M. Y. Yildirim, S. Alcorn, J. He, H. Davulcu, and H. Tong. 2017. Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In *Proceedings of the 2017 SIAM International Conference on Data Mining.* SIAM, 570–578.

[54] L. Zheng, Y. Cheng, and J. He. 2019. Deep multimodality model for multi-task multi-view learning. In *Proceedings of the 2019 SIAM International Conference on Data Mining.* SIAM, 10–18.

[55] D. Zhou, J. He, K S. Candan, and H. Davulcu. 2015. MUVIR: Multi-View Rare Category Detection.. In *IJCAI (2015)*.

[56] D. Zhou, S. Zhang, M. Y. Yildirim, S. Alcorn, H. Tong, H. Davulcu, and J. He. 2017. A local algorithm for structure-preserving graph cut. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 655–664.

[57] Y. Zhou and J. He. 2017. A randomized approach for crowdsourcing in the presence of multiple views. In *2017 IEEE International Conference on Data Mining (ICDM).* IEEE, 685–694.

[58] Y. Zhou, L. Ying, and J. He. 2017. MultiC2: an optimization framework for learning from task and worker dual heterogeneity. In *Proceedings of the 2017 SIAM International Conference on Data Mining.* SIAM, 579–587.

[59] Y. Zhu, J. Li, and J. He. 2017. Learning from multi-modality multi-resolution data: an optimization approach. In *SDM (2017)*.