engineering proceedings

MDPI

Proceeding Paper

Stock Embeddings: Representation Learning for Financial Time Series †

Rian Dolphin [1,*], Barry Smyth [1,2] and Ruihai Dong [1,2]

[1] School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland; barry.smyth@ucd.ie (B.S.); ruihai.dong@ucd.ie (R.D.)
[2] Insight Centre for Data Analytics, University College Dublin, D04 V1W8 Dublin, Ireland
* Correspondence: rian.dolphin@ucdconnect.ie
† Presented at the 9th International Conference on Time Series and Forecasting, Gran Canaria, Spain, 12–14 July 2023.

Abstract: Identifying meaningful and actionable relationships between the price movements of financial assets is a challenging but important problem for many financial tasks, from portfolio optimization to sector classification. However, recent machine learning research often focuses on price forecasting, neglecting the understanding and modelling of asset relationships. To address this, we propose a neural model for training stock embeddings that harnesses the dynamics of historical returns data to reveal the nuanced correlations between financial assets. We describe our approach in detail and discuss several practical ways it can be used in the financial domain. Specifically, we present evaluation results to demonstrate the utility of this approach, compared to several benchmarks, in both portfolio optimization and industry classification.

Keywords: time series; representation learning; asset relationships; risk management; industry classification

check for updates

Citation: Dolphin, R.; Smyth, B.; Dong, R. Stock Embeddings: Representation Learning for Financial Time Series. *Eng. Proc.* **2023**, *39*, 30. https://doi.org/10.3390/engproc2023039030

Academic Editors: Ignacio Rojas, Hector Pomares, Luis Javier Herrera, Fernando Rojas and Olga Valenzuela

Published: 29 June 2023

1. Introduction

The stock market is a challenging but appealing target for time series analysis [1,2], and it has long attracted the attention of researchers. While state-of-the-art time series techniques have demonstrated an excellent performance in the financial domain [3], much of the recent literature applying time series analysis to financial markets overlooks relational information. Instead, assets are analysed in isolation while valuable relational information is overlooked [4]. Understanding asset relationships is essential for several important financial tasks like portfolio optimization, hedging, and sector classification [5]. The conventional measure of asset similarity is correlation, popularised by Markowitz's seminal paper on modern portfolio theory [6]. However, there has been criticism of the application of correlation to financial returns [7] and recently proposed alternative similarity measures include geometric [8] and adjusted correlation-based approaches [9].

In recent years, learning embedding representations have led to breakthroughs in capturing semantic relationships in natural language processing [10]. However, the applications of embeddings in finance are mainly limited to applying pre-trained large language models to textual data, with very limited work on learning embeddings directly from non-textual financial data such as historical returns [11]. For example, the authors in [12–14] use event embeddings from financial news for stock return forecasting, [15] employ BERT in annual report texts, and [16] uses word embeddings for stock selection.

In this paper, we outline a novel methodology that allows for rich relational information to be extracted from financial returns time series and encoded using embedding representations. After a detailed description of the approach in Section 2, we present two evaluations to showcase how the learned representations are useful in tackling the

real-world financial problems of sector classification and portfolio optimization. The main contributions can be summarized as follows:

- A novel approach to learning embedding representations of time series is proposed and applied in the context of financial assets.
- In contrast to existing sector classification schemes, which are highly subjective, we showcase how the learned representations can be used to objectively segment stocks into industry sectors.
- The learned embeddings are used within a novel approach for portfolio construction that results in portfolios with statistically significantly lower out-of-sample volatility.

2. Architecture and Approach

Inspired by the distributional semantics area of natural language processing, the model described in this section uses the idea of *context stocks* to learn the embeddings of target stocks. In linguistics, the distributional hypothesis, which underpins a number of popular language models [10], encodes the idea that words commonly occurring in the same contexts tend to have similar meanings.

In the financial domain, a similar hypothesis also holds: companies with similar characteristics—such as those operating in the same business sectors—tend to exhibit similar stock price fluctuations [17]. By engineering the selection of context stocks to reflect this hypothesis, and adding noise reduction strategies, our proposed framework generates embeddings that capture nuanced relationships between financial assets purely based on historical pricing time series.

2.1. Generating Training Data

Consider a universe of stocks $U = \{a_1, \ldots, a_n\}$. For each stock a_i, there is a time series $\mathbf{p}_{a_i} = \{p_0^{a_i}, \ldots, p_T^{a_i}\}$ representing its price at discrete points in time $t \in \{0, 1, \ldots, T\}$ (daily or weekly, for example). From these pricing data, we can compute a *returns time series* $\mathbf{r}_{a_i} = \{r_1^{a_i}, \ldots, r_T^{a_i}\}$ using Equation (1).

$$r_t^{a_i} = \frac{p_t^{a_i} - p_{t-1}^{a_i}}{p_{t-1}^{a_i}} \tag{1}$$

Using these returns time series, we can generate sets of stocks called *target:context sets* made up of a target stock and its set of context stocks. More concretely, for a context size C (a hyperparameter), the context stocks for the target asset a_i at time t are the C stocks, which have the closest return at that point in time. The closest return is defined by the lowest absolute value difference in return between candidate stock a_j and the target asset a_i, formulated as $\left| r_t^{a_i} - r_t^{a_j} \right|$. An example of this process is outlined in Figure 1, with AAPL as the context stock and t is 3 January 2000. We compute the absolute value difference between the return of AAPL at that point in time with the return of each other stock at the same point in time. Then, we choose the C stocks with the lowest values (most similar) as the context stocks, excluding AAPL itself. In this case, IBM and MSFT have the smallest difference with AAPL and so are chosen as context stocks. We generate a target:context set for every stock at each point in time, which results in a total of $|U| \times T$ sets for training.

An example of a target:context set for $C = 3$ is $\mathcal{S}(a_1, t) = [a_1 : a_{270}, a_{359}, a_{410}]$, which corresponds to *[Apple: IBM, Microsoft, Oracle]* since, for example, 270 is the index value for IBM in the dataset and so a_{270} corresponds to IBM. This tells us that, at a certain point in time t, the three stocks with the closest returns to Apple Inc. were IBM, Microsoft and Oracle.

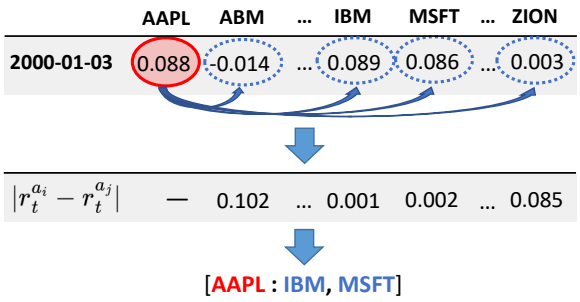

Figure 1. Generating training data, i.e., target:context stock sets.

2.2. Base Model Architecture

The proposed model architecture is illustrated in Figure 2 with *the stock embeddings as the model parameters*. Thus, each row in the weight matrix $\mathbf{W} \in \mathbb{R}^{|U| \times N}$ is a stock embedding, all of which are randomly initialized. Here, N is the embedding size (a hyperparameter), and $|U|$ is the number of stocks/assets in the dataset. The architecture is unusual because the goal of the model is not to make predictions in a downstream task; rather, its sole purpose is to learn parameters such that the resulting embeddings encapsulate the relationships present in the underlying returns time series. In the remainder of this section, the model architecture is described in detail from *input* (the context stocks $a_{j_1}, a_{j_2}, \ldots, a_{j_C}$, where $j_1, j_2, \ldots, j_C \in \{1, 2, \ldots, |U|\}$ are the index values of context stocks) to *output* (the probability of each stock being the associated target).

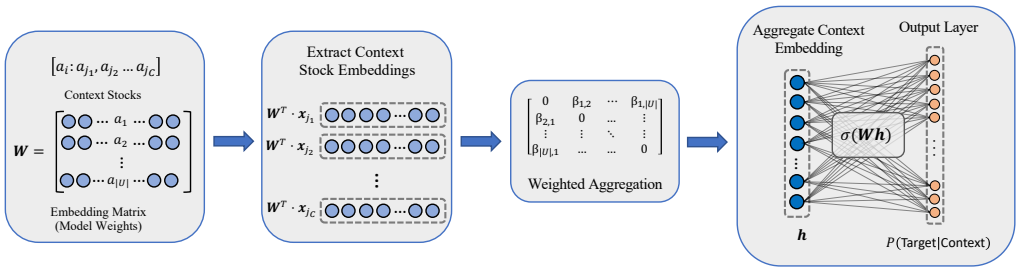

Figure 2. Model Architecture

The first step is to compute the hidden layer $\mathbf{h} \in \mathbb{R}^N$, which can be thought of as an aggregation of the context stocks' embeddings. To do this, each of the context stocks is one-hot encoded, which allows for us to easily extract the relevant embedding (row) from $\mathbf{W}$. For example, the first context stock a_{j_1} is encoded as $\mathbf{x}_{j_1} \in \mathbb{R}^{|U|}$, a one-hot vector of all zeros except for the element in position j_1. As a result, computing $\mathbf{W}^T \cdot \mathbf{x}_{j_1}$ will extract a single row from $\mathbf{W}$— the embedding corresponding to the first context stock a_{j_1}.

In this way, C one-hot vectors are obtained, $\{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \ldots, \mathbf{x}_{j_C}\}$, one for each context stock. The embeddings corresponding to the C context stocks are then extracted by pre-multiplying each one-hot vector by $\mathbf{W}^T$ as previously described. Finally, to compute the hidden layer $\mathbf{h}$, the context stock embeddings are aggregated. The most basic form of aggregation proposed is an element-wise average of the extracted embeddings, which is formalized in Equation (2).

$$\mathbf{h} = \frac{1}{C}\mathbf{W}^T(\mathbf{x}_{j_1} + \mathbf{x}_{j_2} + \cdots + \mathbf{x}_{j_C}) \tag{2}$$

Thus, the hidden layer, $\mathbf{h}$, is an N-dimensional vector and can be thought of as an aggregate embedding representation of the context stocks. In Equation (2), each embedding receives an equal weighting of C^{-1}; however, in Section 2.3, we describe a more complex aggregation approach intended to reduce noise.

The next step is to estimate the conditional probability of a particular stock a_i being the target stock *given* the context stocks, which have aggregate embedding $\mathbf{h}$. This is computed

as shown in Equation (3), where we define $\mathbf{v}_{a_k}^{\mathbf{T}} := \mathbf{W}^{\mathbf{T}} \cdot \mathbf{x}_{a_k}$, interpreted as the embedding corresponding to asset a_k.

$$\mathbb{P}(a_T = a_i \mid a_{j_1}, \ldots, a_{j_C}) = \frac{\exp(\mathbf{v}_{a_i}^{\mathbf{T}} \cdot \mathbf{h})}{\sum_{k=1}^{|U|} \exp(\mathbf{v}_{a_k}^{\mathbf{T}} \cdot \mathbf{h})} \tag{3}$$

Ensured by the softmax activation, the output is a conditional probability expressing the probability stock a_i is the target stock, given the context stocks that were observed. The goal is to learn the weight/embedding matrix that maximizes the conditional probability for the correct target stock a_T. As a result, we frame this as an optimization problem where the goal is minimizing the loss function shown in Equation (4) with respect to $\mathbf{W}$, which can be achieved using stochastic gradient descent.

$$\mathcal{L} = -\mathbf{v}_{a_T}^{\mathbf{T}} \cdot \mathbf{h} + \log \sum_{k=1}^{|U|} \exp(\mathbf{v}_{a_k}^{\mathbf{T}} \cdot \mathbf{h}) \tag{4}$$

In this way, after training, stocks that commonly co-occur in target–context sets will have high similarity representations in the latent space, as desired.

2.3. Noise Reduction Strategies

Financial returns data are notoriously noisy [18], and so, in addition to the base model architecture, we propose two amelioration strategies to improve performance. Firstly, a weighting strategy based on overall distributional co-occurrence is introduced, and included in Figure 2. With this, the hidden layer $\mathbf{h}$ is computed via a weighted average, and is implemented by scaling each $\mathbf{x}_j$ using a weight, which is proportional to the rate at which the given context stock a_j appears in the context of the target stock a_i over every time point in the training dataset ($t = 1, 2, \ldots, T$). This is outlined in Equation (5), where $\mathbb{1}$ denotes the indicator function.

$$w_{i,j,t} \propto \beta_{i,j} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\Big(a_j \in \mathcal{S}(a_i, t)\Big) \tag{5}$$

The constant of proportionality here, $k_{i,t}$, is computed such that the weightings over all context stocks sum to one.

$$w_{i,j,t} = k_{i,t} \cdot \beta_{i,j} \quad : \quad k_{i,t} = \left(\sum_{j : a_j \in \mathcal{S}(a_i,t)} \beta_{i,j} \right)^{-1} \tag{6}$$

Secondly, the distribution of returns over a short time period, such as daily, contains a large proportion of values close to 0, which indicates little movement in stock price. In an effort to isolate meaningful cases and reduce noise, a context set $\mathcal{S}(a_i, t)$ was deleted from the training data if the target stock return, $r_t^{a_i}$, was within the interquartile range (IQR) of returns on that day. As a result, only sets where the target stock had a movement outside the IQR of the market average on a given day were included in training.

3. Evaluation Dataset

In the following two sections, we present the results of initial evaluations of the proposed stock embeddings approach. We describe two experiments to evaluate different aspects of the learned distributed representations. In Section 4, we evaluate the ability of the embeddings to capture sectoral similarity by: visualizing the embeddings in 2D space, examining nearest-neighbour stocks, and finally quantifying the results through sector classification. In Section 5, we describe an evaluation within the portfolio construction setting and show that the proposed approach results in out-of-sample portfolios with statistically significantly lower volatility than conventional hedging strategies.

In both experiments, we used a publicly available dataset of daily pricing data for 611 US stocks during the period 2000–2018. In addition to daily returns, each stock is also associated with a *sector* and *industry* classification label from the Global Industry Classification Scheme (GICS). The former corresponds to the business sector in which the company operates—there are eleven sectors in total, including Finance, Health Care and Technology, for example—while the latter represents a finer-grained classification so that a stock in the Technology sector may have Computer Software as its industry label, for example, to contrast it with another Technology stock in the Electronic Components industry. The embeddings used in both experiments were generated using a context size of 3 and an embedding dimension of 20. Results are reported both with and without noise reduction techniques.

4. Evaluation 1: Sector Classification

Stock prices are influenced by a myriad of hidden factors and unpredictable events, making investing a risky venture. Individual stocks expose investors to both market risk (systematic) and asset-specific risk (idiosyncratic). Exchange-traded funds (ETFs) have been growing in popularity because they allow investors access to mitigate idiosyncratic risk, through diversification, at very low costs. They are securities traded on public exchanges, that provide partial ownership in large portfolios of stocks. These portfolios often track a specific market sector or geographical region, allowing for investors diversified exposure to desired market segments. By 2016, ETFs' market share surpassed 10% of total US market capitalization, accounting for over 30% of trading volume [19].

However, ETF providers must decide on portfolio constituents, which is currently a very subjective process, particularly for ETFs tracking particular market sectors. For instance, a strong case could be made for Amazon, classified as consumer discretionary by the GICS, to be included in consumer discretionary, technology, or consumer staples ETFs.

Aside from deciding on ETF constituents, segmenting stocks into market sectors is also crucial for many other types of financial and economic analysis—measuring economic activity, identifying peers and competitors, quantifying market share and bench-marking company performance—none of which would be possible without industry classifications [5]. A well-defined sector classification system also facilitates relative valuation and sector-specific return and risk estimates [20].

We demonstrate the utility of stock embeddings in classifying stocks into business sectors and identifying inconsistencies in existing classification schemes. We first visualize the latent space of learned embeddings to examine stock clustering and relationships. We then present a nearest neighbors analysis, and finally train a classification model using the embeddings to assign sector labels to companies.

4.1. Clustering Stocks Using Embeddings

Visualizing latent embeddings in a lower dimensional space can often be useful to identify relationships and clustering behavior. Figure 3 shows a graphical representation of the embeddings for stocks in four of the largest business sectors: Energy, Finance, Public Utilities and Technology. Each node represents a stock, colored by sector, and an edge indicates that the two nodes it connects have greater than 0.7 cosine similarity between their embeddings. The plot is generated using a force-directed graph-drawing algorithm.

The clustering of stocks into business sectors is clearly evident in Figure 3. We can also see that nearly all the edges in the graph are between nodes from the same sector. From this, we conclude that the proposed model architecture and training procedure result in embeddings that successfully capture relationships between stocks from their time series.

When we consider that the training data used are purely derived from historical returns data, this is a very positive result because it suggests that it is possible to reconstruct important sectoral information from the embeddings, and indeed this can likely be achieved a way that is more nuanced than might be possible using simple sectoral labels.

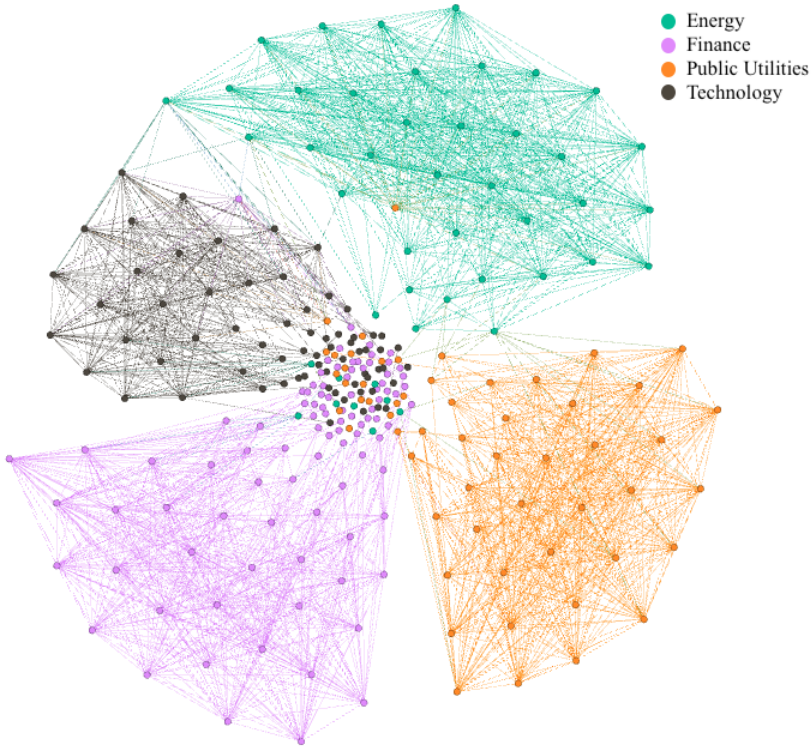

Figure 3. Graphical Visualization of Stock Embeddings Colored by Business Sector.

4.2. Identifying Nearest Neighbor Assets

Ideally, we should expect the embeddings of related stocks to be 'semantically' closer, by some suitable similarity metric, than dissimilar stocks, and the ability to identify similar stocks is an important tool for effective portfolio design. Here, we use cosine similarity as our similarity metric to provide examples of the k nearest neighbors for a sample of example stocks; similar results can be obtained when using alternative metrics such as Euclidean distance.

Table 1 shows the top-3 nearest neighbors for JP Morgan Chase, Analog Devices, and Exxon Mobil Corp, three well-known companies in very different sectors. In each case, the nearest neighbors pass the "sanity test" in that they belong to similar sectors and industries. For example, the three nearest neighbors of JP Morgan, a major bank, are also all major banks. Remember, no sectoral or industry information was used to determine these nearest neighbors, and only daily returns time series were used to generate the embeddings used for similarity assessments.

Table 1. Examples of Top-3 Nearest neighbors for Given Query Stocks.

Query Stock Sector-Industry	3 Nearest Neighbors-Sector-Industry	Similarity
JP Morgan Chase	Bank of America Corp-Finance-Major Bank	0.88
Finance	State Street Corp-Finance-Major Bank	0.82
Major Bank	Wells Fargo & Company-Finance-Major Bank	0.81
Analog Devices	Maxim Integrated-Technology-Semiconductors	0.93
Technology	Texas Instruments-Technology-Semiconductors	0.91
Semiconductors	Xilinx, Inc.-Technology-Semiconductors	0.90
Chevron Corporation	Exxon Mobil-Energy-Oil & Gas	0.89
Energy	BP P.L.C.- Energy-Oil & Gas	0.82
Oil & Gas	Occidental Petroleum-Energy-Oil & Gas	0.78

Nearest neighbor stocks can be used in a variety of ways by investors. By focusing on the k nearest neighbor stocks, we can develop a basic stock recommendation system which, when given a target stock—a novel stock for the investor or one already in their portfolio—can generate a ranked list of similar stocks based on their historical returns data. Conversely, the ability to identify maximally dissimilar stock is an important way to improve portfolio diversity in order to provide an investor with the ability to guard against volatility and *hedge* against sudden sectoral shocks.

4.3. Sector Classification Performance

This section uses the generated embeddings to objectively segment companies into industry sectors. To do this, the learned embeddings serve as the input to a classification model and their corresponding GICS sector labels represent the ground truth. However, classification accuracy is limited by the unpredictable factors inherent in historical returns data and inconsistencies in current subjective approaches to stock labeling [5]. Therefore, sector classification in finance is a challenging task, particularly when relying solely on returns data.

A considerable class imbalance exists among the sectors within the dataset, which can introduce algorithmic bias and adversely affect the results. To mitigate this issue, we implemented the Synthetic Minority Oversampling Technique (SMOTE) [21] on the training data.

Table 2 shows the performance of the proposed methodology in the sector classification task. We also include alternative several time series classification models as baselines. The embedding model with both noise reduction techniques achieves the highest accuracy of 60%. In addition, our method provides the added benefit of producing learned representations that can serve as features in other asset-related tasks.

Table 2. Sector Classification Results.

Model	Precision	Recall	F1	Accuracy
Catch22	0.31	0.35	0.31	35%
Contractable BOSS	0.47	0.39	0.37	39%
RBOSS	0.57	0.42	0.45	42%
Shapelet	0.54	0.42	0.45	42%
Shapelet Transform	0.39	0.46	0.40	46%
WEASEL	0.50	0.47	0.47	47%
MUSE	0.54	0.54	0.51	54%
Time Series Forest Classifier	0.55	0.55	0.53	55%
Canonical Interval Forest	0.57	0.56	0.52	56%
Arsenal	**0.64**	0.58	0.53	58%
Embedding	0.57	0.54	0.55	54%
Embedding + IQR	0.59	0.56	0.56	56%
Embedding + Weight	0.59	0.57	0.57	57%
Embedding + Weight + IQR	0.62	**0.60**	**0.60**	**60%**

This is an impressive result considering the aforementioned limitations and the fact that there are 11 sector classes (Basic Industries, Capital Goods, Consumer Durables, Consumer Non-Durables, Consumer Services, Energy, Finance, Health Care, Public Utilities, Technology, Transportation). A more in-depth analysis shows a variation in accuracy across sectors, with the more populated sectors being very accurately classified (F1 $>$ 0.9 in some cases) while other minority sectors have relatively low accuracy. With this in mind, we believe the accuracy could be improved with a larger dataset.

The objective sector classification approach presented has considerable potential for addressing and mitigating inconsistent company segmentation—a well-documented issue [22]—in practical applications.

5. Evaluation 2: Hedging/Diversification

As previously mentioned, the existing literature on computational methods for stock markets mainly targets returns forecasting. However, not all investors prioritize maximizing returns; for some, portfolio protection is more important. For instance, a defined benefit pension fund portfolio manager primarily focuses on covering agreed benefits, making risk management more crucial than maximizing returns. To ensure protection, investors and portfolio managers use diversification and hedging, measuring effectiveness in terms of volatility reduction. As a result, identifying dissimilar stocks that behave oppositely to similar ones is essential for traders to hedge their target stocks and limit overall risk.

Typically, hedging involves negatively correlated assets and various correlation metrics. We propose an alternative: using generated embeddings to find maximally dissimilar stocks and inform hedging strategies. We evaluate a scenario where an investor holds a position in a stock (query stock) and seeks a single stock (hedge stock) to reduce risk, measured as volatility, as much as possible.

To do this, we create a hedged two-asset long-only portfolio for each stock in the dataset, resulting in 611 portfolios, each containing a query stock and its lowest similarity hedge stock. The similarity is based on cosine similarity between embeddings or baselines from Table 3. We then simulate each portfolio's out-of-sample performance using different similarity metrics, recording realized volatility values. See Algorithm 1.

Algorithm 1 Finding Hedged Portfolios and Simulating Realized Volatility

```
for target_stock in stocks do
    # Initialize variables
    min_similarity ← ∞
    selected_hedge ← None
    # Iterate over all stocks to find the least similar stock as a hedge
    for candidate_hedge in stocks do
        if similarity(target_stock, candidate_hedge) < min_similarity then
            # Update the minimum similarity and the selected hedge stock
            min_similarity ← similarity(target_stock, candidate_hedge)
            selected_hedge ← candidate_hedge
        end
    end
    # Simulate the performance of the hedged portfolio
end
```

Table 3. Portfolio hedging experiment results along with Tukey HSD test indicating significantly lower volatility than Pearson baseline at $\alpha = 0.01$.

Method	Avg Volatility	Significant
Pearson	23.8%	-
Spearman	24.0%	✗
Geometric	23.9%	✗
Embedding	22.9%	✓
Embedding + Weight	22.8%	✓
Embedding + IQR	**21.3%**	✓
Embedding + Weight + IQR	21.9%	✓

A train-test split is crucial to resemble real-world out-of-sample trading applications. Due to financial data's time sensitivity, we use the first 70% of data (2000–2013) for training embeddings and computing baseline similarity metrics, and the remaining 30% (2014–2018) for simulating portfolios and computing realized volatility.

In modern portfolio theory [6], similarity is defined using Pearson correlation, a common method in industry and academia. We suggest that cosine similarity between

proposed embeddings may be superior. The evaluation includes Pearson correlation, Spearman rank-order correlation coefficient, and a recent geometric shape similarity [8] as baselines.

Table 3 displays the average volatility results. To ensure the robustness of results, we reran the experiment 100 times; instead of choosing the single most dissimilar stock as the hedge stock, we randomly chose one of the 25 most dissimilar results for each target stock on each iteration. Overall, the proposed embeddings approach with IQR noise reduction results in portfolios with the lowest average volatility, at 21.3%. Post-hoc Tukey HSD tests indicate that the volatility in all of the embedding-based methods is statistically significantly lower than the Pearson baseline at $\alpha = 0.01$; none of the other baseline approaches generate a significantly lower mean volatility compared to the Pearson approach.

Thus, the proposed embedding methodology can be used to inform a hedging strategy that is superior to a number of baselines, at least within the simplified setting used for this experiment. Obviously, real-world settings are based on more complex portfolios with many different stocks that need to be collectively hedged, and it is a matter for future work to further evaluate our embeddings approach in these more realistic settings. That being said, the approach used here still serves as an important indication of success: had the embeddings approach not been able to demonstrate improved volatility in these simple two-stock portfolios, this would cast doubt on its likely future success in more complex portfolios. Similarly, there is more work to be carried out when it comes to understanding the dynamics of the distributed representations and how they are learned: for example, how changing the context size or embedding dimension impacts these findings.

6. Conclusions

In this paper, we presented a novel methodology for quantifying the relationships between financial assets by learning embedding representations derived solely from historical returns. The effectiveness of the approach is demonstrated through a nearest neighbour case study and benchmark comparisons in two key financial tasks: (1) accurately classifying stocks into their respective industry sectors, and (2) constructing portfolios that exhibit statistically significant reductions in volatility compared to traditional baseline methods.

Moving forward, we aim to further assess the potential of this methodology by examining more intricate portfolio management scenarios and incorporating additional datasets. The proposed technique is versatile and applicable to any group of financial assets with accessible pricing information, allowing us to extend our analysis to encompass multiple asset classes beyond equities. Furthermore, we plan to conduct a thorough exploration of the model parameter space used for learning the embeddings, as well as investigating alternative approaches for generating context stocks.

Author Contributions: Conceptualization, R.D. (Rian Dolphin), R.D. (Ruihai Dong) and B.S.; methodology, R.D. (Rian Dolphin); software, R.D. (Rian Dolphin); validation, R.D. (Rian Dolphin); formal analysis, R.D. (Rian Dolphin); investigation, R.D. (Rian Dolphin); resources, R.D. (Rian Dolphin); data curation, R.D. (Rian Dolphin); writing—original draft preparation, R.D. (Rian Dolphin), and B.S.; writing—review and editing, R.D. (Rian Dolphin), R.D. (Ruihai Dong) and B.S.; visualization, R.D. (Rian Dolphin); supervision, R.D. (Ruihai Dong) and B.S.; project administration, R.D. (Ruihai Dong) and B.S.; funding acquisition, R.D. (Ruihai Dong) and B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All code and data is available at github.com/rian-dolphin/stock-embeddings (accessed on 15 June 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bachelier, L. Théorie de la spéculation. *Ann. Sci. l'École Norm. Supérieure* **1900**, *17*, 21–86. [CrossRef]
2. Fama, E.F. The behavior of stock-market prices. *J. Bus.* **1965**, *38*, 34–105. [CrossRef]
3. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [CrossRef]
4. Sharma, A.; Bhuriya, D.; Singh, U. Survey of stock market prediction using machine learning approach. In Proceedings of the 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbator, India, 20–22 April 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 2, pp. 506–509.
5. Phillips, R.L.; Ormsby, R. Industry classification schemes: An analysis and review. *J. Bus. Financ. Librariansh.* **2016**, *21*, 1–25. [CrossRef]
6. Markowitz, H. *Portfolio Selection*; Wiley: Hoboken, NJ, USA, 1952.
7. Lhabitant, F.S. *Correlation vs. Trends: A Common Misinterpretation*; EDHEC: Roubaix, France, 2020.
8. Chun, S.H.; Ko, Y.W. Geometric Case Based Reasoning for Stock Market Prediction. *Sustainability* **2020**, *12*, 7124. [CrossRef]
9. Dolphin, R.; Smyth, B.; Xu, Y.; Dong, R. Measuring Financial Time Series Similarity with a View to Identifying Profitable Stock Market Opportunities. In Proceedings of the International Conference on Case-Based Reasoning, Salamanca, Spain, 13–16 September 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 64–78.
10. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
11. Dolphin, R.; Smyth, B.; Dong, R. Industry Classification Using a Novel Financial Time-Series Case Representation. *arXiv* **2023**, arXiv:2305.00245.
12. Ding, X.; Zhang, Y.; Liu, T.; Duan, J. Deep learning for event-driven stock prediction. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
13. Ding, X.; Zhang, Y.; Liu, T.; Duan, J. Knowledge-driven event embedding for stock prediction. In Proceedings of the Coling 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2133–2142.
14. Cheng, D.; Yang, F.; Wang, X.; Zhang, Y.; Zhang, L. Knowledge graph-based event embedding framework for financial quantitative investments. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 2221–2230.
15. Ito, T.; Camacho Collados, J.; Sakaji, H.; Schockaert, S. Learning Company Embeddings from Annual Reports for Fine-Grained Industry Characterization. In Proceedings of the Second Workshop on Financial Technology and Natural Language Processing, Kyoto, Japan, 5 January 2020.
16. Hirano, M.; Sakaji, H.; Kimura, S.; Izumi, K.; Matsushima, H.; Nagao, S.; Kato, A. Selection of related stocks using financial text mining. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 191–198.
17. Gopikrishnan, P.; Rosenow, B.; Plerou, V.; Stanley, H.E. Identifying business sectors from stock price fluctuations. *arXiv* **2000**, arXiv:cond-mat/0011145.
18. De Long, J.B.; Shleifer, A.; Summers, L.H.; Waldmann, R.J. Noise trader risk in financial markets. *J. Political Econ.* **1990**, *98*, 703–738. [CrossRef]
19. Ben-David, I.; Franzoni, F.; Moussawi, R. *Exchange Traded Funds (ETFs)*; Technical Report; National Bureau of Economic Research: Cambridge, MA, USA, 2016.
20. S&P.; MSCI. *Global Industry Classification Standard (GICS) Methodology*; MSCI: New York, NY, USA, 2020.
21. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
22. Chan, L.K.; Lakonishok, J.; Swaminathan, B. Industry classifications and return comovement. *Financ. Anal. J.* **2007**, *63*, 56–70. [CrossRef]