Asset Price and Direction Prediction via Deep 2D Transformer and Convolutional Neural Networks

Tuna Tuncer*
Ozyegin University
Istanbul, Turkey
tuna.tuncer@ozu.edu.tr

Uygar Kaya*
Ozyegin University
Istanbul, Turkey
uygar.kaya@ozu.edu.tr

Emre Sefer†
Ozyegin University
Istanbul, Turkey
emre.sefer@ozyegin.edu.tr

Onur Alacam
Ozyegin University
Istanbul, Turkey
onur.alacam@ozu.edu.tr

Tugcan Hoser
Ozyegin University
Istanbul, Turkey
tugcan.hoser@ozu.edu.tr

ABSTRACT

Artificial intelligence-based algorithmic trading has recently started to attract more attention. Among the techniques, deep learning-based methods such as transformers, convolutional neural networks, and patch embedding approaches have become quite popular inside the computer vision researchers. In this research, inspired by the state-of-the-art computer vision methods, we have come up with 2 approaches: DAPP (Deep Attention-based Price Prediction) and DPPP (Deep Patch-based Price Prediction) that are based on vision transformers and patch embedding-based convolutional neural networks respectively to predict asset price and direction from historical price data by capturing the image properties of the historical time-series dataset. Before applying attention-based architecture, we have transformed historical time series price dataset into two-dimensional images by using various number of different technical indicators. Each indicator creates data for a fixed number of days. Thus, we construct two-dimensional images of various dimensions. Then, we use original images valleys and hills to label each image as Hold, Buy, or Sell. We find our trained attention-based models to frequently provide better results for ETFs in comparison to the baseline convolutional architectures in terms of both accuracy and financial analysis metrics during longer testing periods. Our code and processed datasets are available at https://github.com/seferlab/SPDPvCNN

CCS CONCEPTS

• **Computing methodologies → Machine learning**; **Computer vision**; • **Applied computing → Economics**.

KEYWORDS

Stock Price Prediction, Deep Learning, Attention, Transformers, Convolutional Neural Networks

ACM Reference Format:
Tuna Tuncer, Uygar Kaya, Emre Sefer, Onur Alacam, and Tugcan Hoser. 2022. Asset Price and Direction Prediction via Deep 2D Transformer and Convolutional Neural Networks. In *3rd ACM International Conference on AI in Finance (ICAIF '22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3533271.3561738

1 INTRODUCTION

Prediction of asset prices such as stocks via artificial intelligence systems exists for almost the last 30 years. Nowadays, stocks are not the only instruments to be traded by traders and institutional investors. There are many more instruments such as options, Exchange-Traded Funds (ETFs), etc [1]. In line with such increase in the number of instruments, artificial intelligence-based trading systems have become more important and functional in a number of different global markets [1].

Recently, deep learning methods have outperformed more traditional machine learning-based models such as SVMs in a variety of classification or prediction tasks. Image processing tasks such as image segmentation or image classification appear to be one of the main domains where deep learning outperforms these more traditional methods [23]. A number of deep learning techniques have also started to appear for financial prediction and classification tasks such as asset price or direction prediction. Some examples are Long Short Term Memory (LSTM) [10], Convolutional Neural Network (CNN) [3, 24], Recurrent Neural Network (RNN) [16], and Transformers [4]. However, application of these deep learning techniques on financial prediction tasks is not as common as their applications in computer vision. Among these techniques, CNNs have achieved one of the best performances [23] even though CNN architectures have mainly been used to solve computer vision tasks including image classification so far. On the other hand, Transformers [8, 30] have recently been introduced mainly for sequence analysis tasks in natural language processing, but since then they have outperformed CNNs in these vision tasks. For instance, Vision Transformer (ViT) [8] achieves quite promising results and outperforms the best-performing convolutional neural networks, while its cost of training is also remarkably smaller than these state-of-the-art convolutional networks. Transformers apply the attention and self-attention mechanisms where attention is useful in drawing connections among any parts of the sequence or image. Relatedly, self-attention is an attention mechanism which relates

*Both authors contributed equally to this research.
†Corresponding author.

ICAIF '22, November 2–4, 2022, New York, NY, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9376-8/22/11.
https://doi.org/10.1145/3533271.3561738

different parts of an input while computing a representation of the same input. As a result, modeling longer range dependencies do not become a problem since the attention mechanisms introduction.

Here, we come up with two approaches in our algorithmic trading framework which predict both asset prices and directions: DAPP (Deep Attention-based Price Prediction) and DPPP (Deep Patch-based Price Prediction). One of them, DAPP, is a transformer-based algorithm that fully depends on using the attention and self-attention mechanisms. DAPP is based on Vision Transformer (ViT) which is a modified version of traditional transformers for computer vision tasks. Our second approach, DPPP, does not use the attention mechanism, but instead it is based on patch embedding-based convolutional neural networks. Patch-embedding convolution neural networks can be considered between transformers and traditional convolutional neural networks, where they also use the patch embeddings observed in transformers but they do not use the attention mechanism. As the first step, both approaches mainly transform one-dimensional financial time series dataset into a two-dimensional dataset similar to image. Via such dimension expanding transform, we can better incorporate successful vision transformers, CNNs, and patch-embedding qualities and capabilities into algorithmic trading. While transforming into such 2D representation, we use up to 65 technical indicators having various parameter combinations for certain time period to define every column's values. In this case, x axis is made up of historical time-series data for every indicator corresponding to every row. We order the rows of this 2D representation such that indicators exhibiting similar patterns are clustered so that y-axis changes are smooth and consecutive patterns can be captured by the deep learning techniques in DAPP and DPPP.

In our models, DAPP and DPPP, we generate images of various dimensions via technical indicators and input them into vision transformer [8] and patch embedding-based convolution neural network respectively. The number of algorithmic trading approaches via transforming time series dataset into 2D representation is quite limited [24]. CNN-TA proposed in [24] uses convolutional neural networks to understand. However, as we will show in the results as well, convolutional neural networks are not superior to more-recently introduced architectures such as transformers or patch embedding-based CNNs. To our best knowledge, algorithmic trading by transforming time series dataset into 2D representation and then processing such input via transformers or patch embedding-based CNNs similar to 2D image classification tasks has not been used before even for other financial prediction tasks. According to our detailed experiments, both of our approaches achieve the best performance across longer time periods compared to well-known baseline methods and similar method only using simpler CNNs without the transformer architecture. DAPP and DPPP have outperformed many trading algorithms over both longer and shorter testing periods including Buy & Hold strategy (BaH), technical indicator-based trading approaches, Multilayer Perceptron (MLP) neural network, LSTM, and enhanced version of CNN-TA which is an another deep convolutional neural network-based prediction method over image-like data [24]. In general, transformer-based DAPP performs slightly better than DPPP across both accuracy and other financial analysis metrics which again shows the importance of attention-based mechanisms combined with patch embeddings in financial tasks such as stock price and direction prediction. Even though the performance of our proposed approaches are promising, we can further enhance their performance via more detailed hyperparameter optimization and fine-tuning.

The upcoming sections of the paper are organized as follows: Section 2 describes the related work after this introduction section. Our proposed solutions are discussed in section 3, which is followed by discussion on implementation and evaluation in section 4. We analyze the evaluation of the proposed approaches in section 5. Lastly, our conclusions are presented in section 6.

2 RELATED WORK

Traditional machine learning methods have been used extensively to forecast stock markets. Number of studies has focused on applying time-series prediction approaches directly to the financial datasets. Another set of studies have applied fundamental or technical analysis techniques to perform accurately in forecasting stock markets. [1] has published a survey on the whole set of forecasting methods including SVM, Artificial Neural Network (ANN), ensemble approaches, etc. [19, 32] have used ANNs to predict stock index values. [2] has come up with a neural network model to forecast in Taiwan stock index. [12] has compared MLP and dynamic ANN to forecast in US stock market. Additionally, [25] has come up with an ANN model which integrates a number of technical analysis indicators in predicting turning points of stock prices. In another work, [9] has focused on predicting foreign exchange (FX) via genetic algorithms by correcting the estimation errors. Lastly, a number of studies have focused on hybrid models to forecast stock prices. For instance, [31] proposes an approach to combine SVM with Principal Component Analysis (PCA).

Recently, newer deep learning approaches has started to appear as the computational capacity has increased. Deep learning is basically a special case of Artificial Neural Network (ANN) which is made up of more than one layer where each layer contributes differently in way to outperform the shallower neural networks [18]. Various types of deep learning models exist such as Convolutional Neural Network (CNN), Long Short Term memory (LSTM), Recurrent Neural Networks (RNN), and Transformers. These deep learning approaches are utilized across various domains. For instance, CNNs are highly utilized in classifying and recognizing the images [14, 17]. Additionally, CNNs also appear frequently in video processing as well as natural language processing tasks such as sentence categorization [13]. LSTMs and RNNs are mainly used to analyze sequential datasets appearing mostly in speech processing, natural language processing, and time-series related tasks. Lastly, more recently-introduced transformers are also used to analyze the sequential datasets. They have first been applied to NLP tasks [33], but recently they have also been successfully applied to computer vision tasks [8, 15].

Deep learning methods, especially deep CNNs and transformer-based architectures are the most commonly-used methods in the last decade. However, number of deep learning approaches developed for financial tasks is quite limited. For instance, [7] has come up with a deep learning-based approach to predict stock markets by using events data by extracting news and other textual knowledge from web. In their paper, they have focused on using a neural tensor network and deeper CNN to take into account the shorter and longer horizon effects of events on stock price fluctuations across US stock market. Another paper [22] has compiled the set of existing deep learning methods to analyze financial time-series datasets such as stock market index prices via methods including convolution and pooling, RNN, autoencoder, etc. [10] has forecasted the stock trends and directions in S&P 500. Similarly, [16] has evaluated random forests and deep neural networks in forecasting the stock prices across S&P 500. Moreover, [34] has developed an approach to forecast the stock price trends by integrating news on Japanese exchange, which has outperformed more traditional machine learning-based method SVM. Lastly, [26] and [24] are more relevant work to ours. In [26], they integrate technical analysis indicators into their prediction. They have developed evolutionary methods to select the optimal technical analysis parameters for indicators, and afterwards used a feedforward neural network (FNN) that uses those optimal parameters as the input. Similarly, [24] come up with an approach called CNN-TA, which combines CNNs with two-dimensional matrix characterization of the technical analysis datasets. According to the both relevant papers, deep learning can learn and generalize quite accurately across buy and sell time points over longer testing horizons.

Similarly, number of transformer-based approaches for prediction tasks across financial markets is extremely limited. For instance, [36] proposes a novel transformer architecture that is based on accurate description via smaller sample feature engineering to capture financial datasets temporal relationships. In another work, [6] proposes a novel transformer-based method to predict stock movements. Their approach integrates a multi-scale Gaussian prior to further improve the transformer's locality. Lastly, [20] comes up with a capsule network that is based on transformer in extracting deeper semantic attributes of the social media dataset's rich semantics.

The existing studies use CNNs quite frequently for image analysis and classification tasks, while CNNS are not mainly used for time-series datasets. CNNs are quite successful on computer vision tasks and the performance increase achieved by CNNs are significant. Deep learning methods, more frequently LSTM and RNN, have recently been the common implementation options for financial time-series prediction. On the other hand, the existing algorithmic trading approaches frequently use technical analysis indicators together with different input. Nevertheless, there are not many methods which combined such technical analysis datasets with deep learning techniques. Furthermore, there are only few studies in algorithmic trading that combine CNNs with two-dimensional matrix characterization of the technical analysis datasets [5, 23–25]. In these studies, deeper CNN models are integrated with the technical analysis datasets. These methods typically convert financial time-series prediction problem into an image classification task by generating two-dimensional images of indicators from the prices.

3 OUR PROPOSED METHODS: DAPP AND DPPP

Following subsections 3.1-3.3 are common for both methods. Sections 3.4-3.5 provide specific details for DAPP and DPPP respectively.

3.1 Dataset Preparation

To analyze financial data, fundamental analysis and technical analysis (TA) are two frequently-used techniques [1]. Fundamental analysis examines financial data specific to a company such as return on equity, cash flow, and balance sheet. On the other hand, technical analysis examines the historical time-series dataset via mathematical models. Number of available technical indicators to predict the financial asset prices are quite high.

In our case, Open-High-Low-Close-Volume (OHLCV) data for 9 of the selected Exchange-Traded Funds (ETFs) were collected from finance.yahoo.com via the yfinance library for training and testing intention. These ETFs are XLF, XLU, QQQ, SPY, XLP, EWZ, EWH, XLY, XLE. We chose those ETFSs since they have data trading data for large number of days with a high volume. For each of these ETF, 65 different technical indicators [21] with different time horizons based on various categories including overlap studies, momentum, volume, volatility, price transformation, and statistics, were calculated by using OHLCV data and TA-Lib (Technical Analysis Library) in Python. Table 1 shows all technical indicators used in our analysis (See https://mrjbq7.github.io/ta-lib/ for more details for these indicators). We have used ETF prices over 20 year period, from 1/1/2002 to 1/1/2022 for training and testing our approaches. This 20 year period covers two major crises: 2008 economic crisis and 2020 COVID crisis.

Table 1: Technical Analysis Indicators Used in DAPP and DPPP

Category	Indicators
Overlap Studies	BBANDSL, BBANDSM, BBANDSU, DEMA, EMA, HT_TRENDLINE, KAMA, MA, MIDPOINT, MIDPRICE, SMA, TEMA, TRIMA, WMA
Momentum	ADX, ADXR, APO, AROONUP, AROONDOWN, AROONOSC, BOP, CCI, CMO, DX, FASTD, FASTDRSI, FASTK, FASTKRSI, MACD, MACDEXT, MACDFIX, MFI, MINUS_DI, MINUS_DM, MOM, PLUS_DI, PLUS_DM, PPO, ROC, ROCP, ROCR, ROCR100, RSI, SLOWD, SLOWK, TRIX, ULTOSC, WILLR
Volume	AD, ADOSC, OBV
Volatility	TRANGE
Price Transformation	AVGPRICE, MEDPRICE, TYPPRICE, WCLPRICE
Statistics	BETA, CORREL, LINEARREG, LINEARREG_ANGLE, LINEARREG_INTERCEPT, LINEARREG_SLOPE, STD, TSF, VAR

3.2 Labeling

In the labeling phase, all daily closing prices of the images are labeled as *Hold*, *Buy*, or *Sell* according to threshold value, after extracting the data for the intended period. The portion below the threshold points are labeled as *Buy*, the portion above the threshold points are labeled as *Sell*, and the remaining points are labeled as *Hold*. The selection of the threshold value is very crucial since the frequency of the generated datasets (such as high-frequency) depend on the threshold value. In our case, by choosing two specific threshold values 0.0038 and 0.01, we generate both a balanced and an imbalanced datasets to be analyzed. We test the financial quality and the robustness of our proposed solutions by evaluating them over both balanced and imbalanced datasets. With a threshold of 0.0038, the data is distributed as evenly as possible so that forecasts are made over the balanced dataset. On the other hand, at a threshold of 0.01, it is a good idea to buy ahead if the asset price increases by 1% in a day, and sell if it drops by 1%. In this case, generating the datasets by using a threshold of 0.0038 establishes the framework for a higher-frequency trading approach. But, threshold of 0.01 makes forecasts that are more hold-centric and are comparable to a buy-and-hold strategy.

3.3 Technical Indicators to Images

We transformed our time-series datasets into 2D images to utilize in our methods and some of the baseline approaches. In the image creation phase, as discussed above, for each day, RSI, WMA, EMA, SMA, ROC, CMO, CCI, PPO, TEMA, WILLR, MACD, and 54 more technical indicator values for different horizons are calculated using the TA-Lib library (These indicator horizons range from 6 to 20 days). These indicators can also be seen as financial time-series filters which are used mostly for medium term algorithmic trading which is our main focus in this research. Different set of indicators with longer horizon parameters can be used for models aiming for longer term algorithmic trading.

Since we have generated our images using these 65 different technical indicators for 65 historical days, the size of the images we obtained is 65 × 65. In addition, each row in the images we have generated shows 65 different technical indicators, while each column represents a different previous day. The order of the indicators is important since different orderings will result in different image formations. The resulting indicator values were sorted by their categories and correlation values, and images were generated using that ordering. In addition, after the images were generated, the standardization process was applied to each image in order to make each of these images more consistent. As an example, Figure 1 illustrates sample 65 × 65 pixel images that are generated during the image generation and labeling phases.

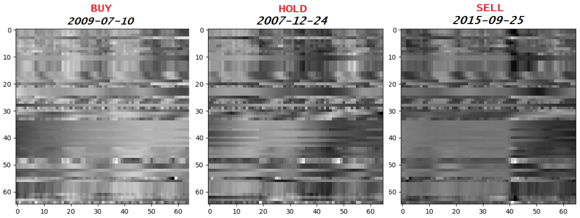

Figure 1: 65 × 65 **Pixel Labeled Sample Images**

3.4 Deep Attention-based Price Prediction: DAPP

Once data collection, labeling, and image conversion steps are accomplished, DAPP applies Vision Transformer (ViT) [8] by training with the generated images and constructed labels. During the adaptation and development of transformers with self-attention-based architectures for computer vision tasks over images, one of the main challenges is that the run time of the model becomes quadratic with respect to the number of pixels, as each pixel will attend to one another. Vision Transformer solved this problem by dividing the image into patches and applying self-attention to each patch separately. Meanwhile, Vision Transformer flattens each patch and applies positional embedding on top, before feeding the patches into the transformer encoder structure, thus aiming to preserve the positional information in the images. The Transformer encoder [30] is made up of multi-head self-attention and MLP blocks. Layer normalization is applied before every block, and residual connections are applied after every block. Finally, a single hidden layer MLP is connected to the end of the transformer encoder for classification.

More formally, let C be the number of channels and (H, W) be the resolution of the original image, DAPP reshapes the generated image $x \in \mathbb{R}^{H \times W \times C}$ into a series of flattened two-dimensional patches $x_p \in \mathbb{R}^{N \times P^2 \cdot C}$ where (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ is the resulting number of patches, which also serves as the effective input sequence length for the transformer. The transformer utilizes constant hidden vector size D through all of its layers, so we flatten the patches and map to D dimensions with a trainable linear projection. The output of this projection is referred as patch embeddings.

As indicated in [8], especially with large scale datasets and pre-training, the vision transformer approaches frequently surpass state-of-the-art performing CNNs. However, they also point out that when trained on medium-sized datasets without a strong regularization, Vision Transformer falls a few percentages below state-of-the-art CNN architectures like ResNet. This is thought to be because Vision Transformer lacks the inductive bias that is common in CNNs.

Since our vision transformer-based architecture DAPP is trained on medium-sized datasets (approximately 45000 images), we have also employed Sharpness-Aware Minimization (SAM) technique [11] to enhance the quality of our results and make them more robust. Number of heads in Vision Transformer part of DAPP is 4, patch size is 8, projection dimension is 64, and there are 8 transformer layers in DAPP. Unless stated otherwise, batch size for DAPP training is 128, training is run for 100 epochs, and AdaDelta [35] is used as the optimizer which is a stochastic optimizer allowing for per-dimension learning rate for stochastic gradient descent.

3.5 Deep Patch-based Price Prediction: DPPP

Once data collection, labeling, and image conversion steps are accomplished, DPPP applies patch embedding-based ConvMixer [29] by training with the generated images and constructed labels. Recently-proposed ConvMixer architecture has claimed that the main reason for the Vision Transformer's success might be the use of patches in the input representation, rather than the transformer architecture (i.e. the self-attention mechanism). As a result, ConvMixer basically replaces the encoder architecture in Vision Transformer with convolutional neural networks. ConvMixer divides the images into patches, flattens them, applies positional embedding, as in Vision Transformer, and then employs standard convolutions instead of inserting patch embeddings into the transformer encoder. Additionally, the concept of mixing introduced in [28] is another significant component utilized in ConvMixer. Concept of mixing is applied to incorporate the self-attention mechanism's characteristic of mixing distant spatial locations in the ConvMixer architecture by employing significantly larger kernel sizes than usual. In other words, ConvMixer aims to replicate the effects of self-attention even if it does not implement self-attention.

In DPPP, ConvMixer architecture utilizes "tensor layout" patch embeddings with patch size 2 for locality preservation, and then keeps applying $d = 8$ copies of a simple fully-convolutional block which mainly consists of large-kernel with size 7 depthwise convolution, Gaussian Error Linear Unit (GELU) activation, and batch normalization. This repeated fully-convolutional blocks are followed by pointwise convolution, before finishing with global pooling and a simple linear classifier. Unless stated otherwise, batch size for DPPP training is 32, training is run for 200 epochs, and AdaDelta [35] is used as the optimizer.

4 EXPERIMENTAL SETUP

4.1 Data Preparation

As we have performed a financial evaluation on the trained models, we have chosen the test dataset in a continuous interval. We have used ETF prices over 20 year period between 1/1/2002 to 1/1/2022 for training and testing our approaches. We have implemented a sliding window with retraining technique where we choose a consecutive 5 year interval as the training period and choose the following one year as the testing period. Afterwards, train and test periods are shifted one year forward, trained our models again, and tested with the next year. Therefore, we test each year between 2007 and 2021 once via our repeated retraining approach.

The resulting indicator values were sorted by their categories and correlation values, and images were generated using that ordering. In addition, after the images were generated, the standardization process was applied to each image in order to make each of our images more consistent. Given that the images are generated at daily intervals, a total of 45090 images, or 5010 images for each ETF, were generated. All generated images were gathered under a single dataset to train a single model because we didn't have a large enough dataset to build a separate model for each ETF.

4.2 Baseline Approaches: Buy & Hold and Enhanced CNN-TA

One of our basic baseline is Buy & Hold Strategy (BaH) which buys the asset when the test period starts and sells it once the test period is over. We have also employed CNN-TA [24] as our next baseline after enhancing it further. CNN-TA is a deep convolutional neural network-based prediction method over image-like data [24]. It consists of an input layer, two convolutional layers with 3 × 3 filters, a max pooling layer, two dropout layers with rates 0, 25 and 0.5, a fully connected layer, and finally an output layer. CNN-TA is a fully convolutional structure without any patch embeddings and attention mechanisms. The adapted CNN structure of CNN-TA is similar to deep convolutional neural networks utilized in MNIST algorithm, where 28 × 28 images have been used in MNIST as input. In our case, we do not directly apply CNN-TA, but instead increase the number of used indicators further enhancing its performance. So, that is why we name the method as *Enhanced CNN-TA*. We have re-implemented CNN-TA by enhancing it in the following way: We have used 65 × 65 images instead of 15 × 15 image as input, increasing the number of technical indicators used as well as the used data history.

4.3 Performance Evaluation

We evaluate the performance of our methods DAPP and DPPP, baselines enhanced CNN-TA and Buy & Hold strategy by both traditional prediction metrics as well as financial metrics. In terms of computational prediction, we report the performance by using Accuracy, Recall, Precision, and F1 score which assess how well the methods perform the Buy, Hold, and Sell classification. On the other hand, for financial evaluation, daily trading transactions have been simulated according to the models predictions, using the test dataset. The model's annualized return as well as the annualized risk of the returns during the test period are used to calculate Sharpe Ratio [27], which is one of the most common metric to financially evaluate the performance of a strategy. Sharpe ratio is calculated as the ratio of annualized portfolio return relative to risk free return divided by annualized standard deviation, where we use 3-month US treasury bill return for risk free rate.

Once algorithmic trading is executed for the test periods, we analyze the generated transactions via financial evaluation techniques. We buy, sell, and hold each asset depending on the predicted label. If the predicted label for an asset is Buy and we have not bought the asset previously, we buy the asset with all of the existing capital. We do not take any action if the predicted label is Hold. Lastly, if the predicted label is Sell, we sell the asset at that price if it has been already bought. We ignore the consecutively repeating labels until predicted label becomes different. We assume a trading commission of $1 for each transaction since the traded ETFs are quite liquid, and have $10000 capital at the beginning. Our code and processed datasets are available at https://github.com/seferlab/SPDPvCNN.

5 RESULTS AND DISCUSSION

We use two different evaluation criterion to evaluate the general performance of our models DAPP and DPPP with respect to the baselines: Machine Learning (ML)-based Evaluation and Financial Evaluation. With the ML-based evaluation, metrics such as Accuracy, Recall, Precision, and F1 score were utilized to assess how well the neural network performs the Buy, Hold, and Sell classification, while for Financial Evaluation, daily trading transactions were simulated according to the model's predictions, using the test dataset with a different time intervals which include the recent COVID-19 pandemic period as well. The return produced by the model's choices, as well as the risk involved, are assessed as a result of the simulation.

5.1 Machine Learning-based Evaluation

We analyze the prediction abilities of our models DAPP and DPPP in terms of multiple metrics as seen in Tables 2-3 respectively. Basically, these tables report the classification performances over imbalanced dataset formed by using threshold 0.01. Results are reported in terms of recall, precision, F1 for each price direction category. We also report the overall precision, F1 and accuracy. According to tables, DPPP and DAPP perform almost similarly, and obtain an overall near $0.62 - 0.63$ accuracy. When we analyze the performance of both methods in detail for predicting each class Buy, Sell, and Hold, both methods also perform similarly. In our case, our financial image dataset is still relatively smaller compared to the datasets in other deep learning applications. So, due to such relatively smaller dataset, we believe attention-based transformer architecture DAPP does not fully reach its best performance. We believe the difference between DAPP and DPPP will become more apparent when we train our methods with more data. Additionally, DPPP's performance being almost as good as DAPP across many metrics shows that efficient and careful design of an architecture using patch embedding together with convolutional structures may unexpectedly perform as good as transformer structure. Moreover, due to the imbalanced dataset, it becomes very difficult for the models to distinguish between Buy and Sell labeled images from Hold labeled images. As a result, in most cases across imbalanced dataset, this results in our models predicting Hold for most images.

Table 2: Classification performance of DAPP over imbalanced dataset formed by using threshold 0.01. **Results are reported in terms of recall, precision, F1 for each price direction category. We also report the overall precision, F1 and accuracy.**

	Buy	Hold	Sell
Recall	0.04	0.98	0.01
Precision	0.39	0.63	0.23
F1 Score	0.07	0.77	0.03
Weighted Precision	0.5131		
Weighted F1	0.4995		
Accuracy	0.6235		

Prediction results are slightly lower over the balanced dataset which is formed by using threshold 0.0038, even though we do not show these results due to space limitations. While DAPP and DPPP have better test accuracy in the imbalanced dataset, both methods give worse results for the Buy and Sell labels than in the balanced dataset. We think that this issue arises because the number of Buy and Sell labels in the imbalanced training dataset are lower than in the balanced dataset. In general, when compared to the performance of other studies focusing on stocks rather than ETFs, overall accuracy of our ETF results are better. Such results can be mainly because ETFs are less sensitive to market events, political changes, and economic crisis compared to stocks. As a result, ETFs are more stable and less volatile. This lower volatility results in a better environment for algorithmic trading methods to learn the trading model more easily.

Table 3: Classification performance of DPPP over imbalanced dataset formed by using threshold 0.01. **Results are reported in terms of recall, precision, F1 for each price direction category. We also report the overall precision, F1 and accuracy.**

	Buy	Hold	Sell
Recall	0.07	0.98	0.02
Precision	0.43	0.64	0.28
F1 Score	0.12	0.78	0.04
Weighted Precision	0.5362		
Weighted F1	0.5163		
Accuracy	0.6299		

5.2 Financial Evaluation

We have also evaluated the performance more realistically by financial analysis. We mainly analyze annualized returns for each ETF, as well as Sharpe ratio [27] for equal-weighted portfolio of all these ETFs over different methods. As discussed in Section 4.3, we have simulated financial analysis for each ETF by simulating the transactions over the test set period.

Even though Buy & Hold (BaH) strategy looks like a simpler strategy, it is still a challenging task to beat Buy & Hold strategy in a longer test period. In our case, DAPP's and DPPP's annualized returns have outperformed BaH strategy in 8 out of 9 ETFs during the test period in imbalanced dataset as seen in Table 4 where the highest annualized returns are shown in bold. For instance, DAPP's annualized return is 18.04% for XLF whereas BaH's annualized return for the same ETF is 8.71%, where the difference is more than 2 folds. For some ETFs such as EWH, BaH can result in negative annualized returns whereas adapting a better architecture results in positive returns, and makes the performance even more robust. When BaH strategy has beaten our methods in QQQ, the performance difference between DAPP and BaH is less than 1%. In general, DAPP is the top performer across almost all ETFs. Even though we have used the enhanced version of CNN-TA as our baseline, integrating the transformer architecture as well as patch embeddings have remarkably outperformed Enhanced CNN-TA as seen in Table 4. Enhanced CNN-TA results show the importance of convolutional architectures, but also show that their results can be further enhanced by adapting more recent better-performing techniques. In terms of Sharpe Ratio of the equal-weighted portfolio among these ETFs, our attention-based approach DAPP still performs better than DPPP and the other competing approaches. In this imbalanced dataset, DAPP achieves a Sharpe Ratio of 2.74 which is reasonably high for a strategy that trades equity indices.

Similarly, the annualized returns of DAPP and DPPP have outperformed BaH strategy in 8 out of 9 ETFs during the test period in the balanced dataset as seen in Table 5 where the highest annualized returns are shown in bold. In general, DAPP is the top performing method over almost all ETFs. Integrating transformer architecture as well as patch embeddings into asset price prediction have again remarkably outperformed Enhanced CNN-TA as seen in Table 5 for most ETFs. Even though the annualized returns of individual

Table 4: Financial performance comparison of our methods DAPP and DPPP with respect to Enhanced CNN-TA approach and baseline Boy & Hold strategy for each ETF in imbalanced dataset that is generated at threshold 0.01. **Performance is measured in terms of annualized return for each ETF.**

ETFs	DAPP	DPPP	Enhanced CNN-TA	BaH
XLF	**18.04%**.	16.98%	16.95%	8.71%
XLU	**8.45%**	7.96%	6.67%	4.52 %
QQQ	18.73%	17.38%	16.17%	**19.64%**
SPY	**12.67%**	10.81%	11.45%	7.14%
XLP	**19.78%**	16.59%	18.61%	14.34%
EWZ	**17.74%**	12.99%	14.53%	-2.83%
EWH	3.92%	3.44%	**5.35%**	-3.27%
XLY	**10.68%**	9.41%	9.98%	9.53%
XLE	10.27%	**15.14%**	14.37%	2.27%
Sharpe Ratio	**2.74**	2.47	2.27	1.07

ETFs are still reasonably high, Sharpe ratios become smaller mainly due to increasing return volatility as measured by the standard deviation. For instance, in this balanced dataset, DAPP achieves a Sharpe ratio of 1.82 which is lower than 2.74 ratio obtained over the imbalanced dataset. According to our machine learning-based and financial evaluations so far, both DAPP and DPPP seem like safer strategies than only convolutional architectures in terms of investing into ETFs.

Table 5: Financial performance comparison of our methods DAPP and DPPP with respect to Enhanced CNN-TA approach and baseline Boy & Hold strategy for each ETF in balanced dataset that is generated at threshold 0.0038. **Performance is measured in terms of annualized return for each ETF.**

ETFs	DAPP	DPPP	Enhanced CNN-TA	BaH
XLF	7.99%	4.57%	**9.64%**	3.71%
XLU	**19.65%**	16.18%	14.83%	12.52 %
QQQ	**27.28%**	24.34%	20.60%	16.64%
SPY	**18.24%**	10.18%	11.59%	**7.14%**
XLP	**19.84%**	9.02%	10.51%	14.34%
EWZ	**4.81%**	-1.85%	-2.58%	-2.83%
EWH	3.59%	**6.42%**	0.29%	-3.27%
XLY	12.95%	**15.15%**	12.90%	10.53%
XLE	**17.25%**	13.40%	13.92%	12.27%
Sharpe Ratio	**1.82**	1.52	1.35	1.07

Figure 2 shows our proposed methods capital allocation for XLP and XLE ETFs respectively over imbalanced dataset during shorter test period between 2018 and middle of 2022. As discussed in Section, we start our analysis with $10000 capital and simulate transaction-based financial analysis. In each figure, DAPP and DPPP performances are compared against the other methods during the selected period. Capital allocation results for remaining ETFs are also similar. For most of the test period intervals, trades generated via our proposed methods are quite successful. We have observed different market conditions during our lengthy test period which includes stationary market condition, bull markets, bear markets, etc. However, these different market conditions and fluctuations associated with them have not impacted the trading performance of the proposed methods. As a result, both DAPP and DPPP could lead to high profits even under bear market conditions.

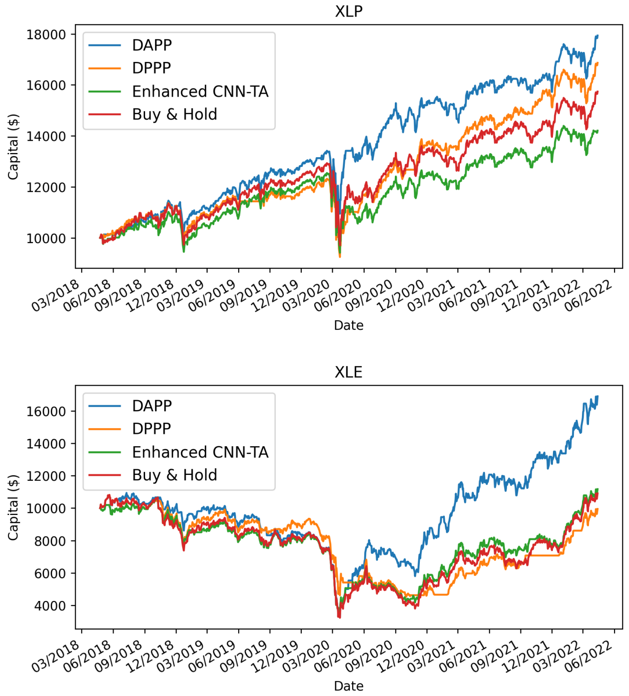

Figure 2: Comparison of our proposed methods DAPP and DPPP with respect to Enhanced CNN-TA and Buy & Hold strategies for XLP and XLE ETFs respectively over the imbalanced dataset.

6 CONCLUSION

In our study, we have developed two methods that uses two-dimensional deep attention-based neural networks and two-dimensional deep patch embedding based convolutional neural networks in predicting financial asset prices. Our approaches utilize a number of technical analysis indicators, and develop algorithmic trading strategies as the end product. By analyzing financial time-series ETF datasets, we have first transformed such data into 2D images as input to our methods DAPP and DPPP. To generate profitable trades, we have focused on forecasting entry and exit points of the time-series values in terms of three categories: Buy, Sell, and Hold. According to our machine learning-based and financial results, our methods perform quite better than Buy & Hold baseline and enhanced version of the only convolutional CNN-TA method over longer out-of-sample test periods. Both the attention mechanism and patch embeddings are useful in increasing the asset price and direction prediction performance.

Even though our methods performance is reasonable, we can still achieve more improvements. In terms of future work, we will first integrate more stocks and ETFs to generate significantly larger amount of data for training our proposed deep learning models. Additionally, we will focus on better analyzing the relationships among technical indicators so that we will generate better 2D representations for prediction tasks. As a result, such better representation will possibly lead to trading models with higher profitability. We will also focus more on optimizing DAPP and DPPP parameters which was not our main focus in this paper. Lastly, we have evaluated the performance by using a long-only strategy. Instead, we may adapt a long-short strategy to remarkably increase the profit as such long-short strategy will decrease the number of times our method will be holding a cash while waiting for a buy or sell signal.

REFERENCES

[1] Rodolfo C. Cavalcante, Rodrigo C. Brasileiro, Victor L.F. Souza, Jarley P. Nobrega, and Adriano L.I. Oliveira. 2016. Computational Intelligence and Financial Markets: A Survey and Future Directions. *Expert Systems with Applications* 55 (2016), 194–211.

[2] An-Sing Chen, Mark T. Leung, and Hazem Daouk. 2003. Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. *Computers & Operations Research* 30, 6 (2003), 901–923. Operation Research in Emerging Economics.

[3] Jou-Fan Chen, Wei-Lun Chen, Chun-Ping Huang, Szu-Hao Huang, and An-Pin Chen. 2016. Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. 87–92.

[4] Siew Ann Cheong, Yawei Li, Shuqi Lv, Xinghua Liu, and Qiuyue Zhang. 2022. Incorporating Transformers and Attention Networks for Stock Movement Prediction. *Complexity* 2022 (2022), 7739087.

[5] Naftali Cohen, Tucker Balch, and Manuela Veloso. 2019. Trading via Image Classification. *CoRR* abs/1907.10046 (2019). arXiv:1907.10046 http://arxiv.org/abs/1907.10046

[6] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. 2020. Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4640–4646. Special Track on AI in FinTech.

[7] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep Learning for Event-Driven Stock Prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) *(IJCAI'15)*. AAAI Press, 2327–2333.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR* abs/2010.11929 (2020). arXiv:2010.11929

[9] Cain Evans, Konstantinos Pappas, and Fatos Xhafa. 2013. Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation. *Mathematical and Computer Modelling* 58, 5 (2013), 1249–1266.

[10] Thomas Fischer and Christopher Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* 270, 2 (2018), 654–669.

[11] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-Aware Minimization for Efficiently Improving Generalization. *CoRR* abs/2010.01412 (2020). arXiv:2010.01412

[12] Erkam Guresen, Gulgun Kayakutlu, and Tugrul U. Daim. 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications* 38, 8 (2011), 10389–10397.

[13] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 655–665.

[14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1725–1732.

[15] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2021. Transformers in Vision: A Survey. *ACM Comput. Surv.* (dec 2021).

[16] Christopher Krauss, Xuan Anh Do, and Nicolas Huck. 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research* 259, 2 (2017), 689–702.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444. https://doi.org/10.1038/nature14539

[19] Zhe Liao and Jun Wang. 2010. Forecasting model of global stock index by stochastic time effective neural network. *Expert Systems with Applications* 37, 1 (2010), 834–841.

[20] Jintao Liu, Hongfei Lin, Xikai Liu, Bo Xu, Yuqi Ren, Yufeng Diao, and Liang Yang. 2019. Transformer-Based Capsule Network For Stock Movement Prediction. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*. Macao, China, 66–73.

[21] Andrew W. Lo, Harry Mamaysky, and Jiang Wang. 2000. Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *The Journal of Finance* 55, 4 (2000), 1705–1765.

[22] Martin Längkvist, Lars Karlsson, and Amy Loutfi. 2014. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters* 42 (2014), 11–24.

[23] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. 2020. Deep learning for financial applications : A survey. *Applied Soft Computing* 93 (2020), 106384.

[24] Omer Berat Sezer and Ahmet Murat Ozbayoglu. 2018. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing* 70 (2018), 525–538.

[25] Omer Berat Sezer, A. Murat Ozbayoglu, and Erdogan Dogdu. 2017. An Artificial Neural Network-Based Stock Trading System Using Technical Analysis and Big Data Framework. In *Proceedings of the SouthEast Conference* (Kennesaw, GA, USA) *(ACM SE '17)*. Association for Computing Machinery, New York, NY, USA, 223–226.

[26] Omer Berat Sezer, Murat Ozbayoglu, and Erdogan Dogdu. 2017. A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters. *Procedia Computer Science* 114 (2017), 473–480. Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA.

[27] William F. Sharpe. 1966. Mutual Fund Performance. *The Journal of Business* 39, 1 (1966), 119–138.

[28] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. 2021. MLP-Mixer: An all-MLP Architecture for Vision. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 24261–24272.

[29] Asher Trockman and J. Zico Kolter. 2022. Patches Are All You Need? *CoRR* abs/2201.09792 (2022). arXiv:2201.09792

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[31] Jianzhou Wang, Ru Hou, Chen Wang, and Lin Shen. 2016. Improved v -Support vector regression model based on variable selection and brain storm optimization for stock price forecasting. *Applied Soft Computing* 49 (2016), 164–178.

[32] Jian-Zhou Wang, Ju-Jie Wang, Zhe-George Zhang, and Shu-Po Guo. 2011. Forecasting stock indices with back propagation neural network. *Expert Systems with Applications* 38, 11 (2011), 14346–14355.

[33] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45.

[34] Akira Yoshihara, Kazuki Fujikawa, Kazuhiro Seki, and Kuniaki Uehara. 2014. Predicting Stock Market Trends by Recurrent Deep Neural Networks. In *PRICAI 2014: Trends in Artificial Intelligence*, Duc-Nghia Pham and Seong-Bae Park (Eds.). Springer International Publishing, Cham, 759–769.

[35] Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR* abs/1212.5701 (2012). arXiv:1212.5701

[36] Qiuyue Zhang, Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang, and Peide Liu. 2022. Transformer-based attention network for stock movement prediction. *Expert Systems with Applications* 202 (2022), 117239.