Stock price prediction using GAN and BERT algorithms. The case of game industry companies.

Jakub Wujec

June 4, 2022

Abstract

In this article the use of the Generative adversarial networks (GAN) model to predict stock market behavior has been proposed. Additionally, an investment strategy that uses predictions from the model has been created. The Technical Analysis and Sentiment Analysis indicators are used as explanatory variables for the model. The study was conducted on four companies from the game industry - Electronic Arts, Ubisoft, Take-Two Interactive Software, and Activision Blizzard. This industry was chosen because of the potential impact of a company's customer reviews on its valuation. The impact is investigated by analyzing the sentiment of selected comments containing prepared keywords. The sentiment is calculated using created by Google NLP model - BERT. GAN tries to predict the future valuation of a given company based on the aforementioned predictors. The price forecast is then used to provide buy or sell signals. For investment strategies in the study, for 3 out of 4 companies it was possible to create a strategy that significantly outperforms the Buy and Hold approach. The main finding of the research is the confirmation of the possibility to use GAN networks to create an effective system that allows us to outperform the Buy and Hold strategy and make a profit despite the fall in the price of an asset.

1 Introduction

In recent years, the stock market has become a place of fierce competition for the best model to predict future prices and consequently, earn money (Prasad, 2021). A significant influence on this has the facilitation of access to computing power, allowing the use of algorithms such as neural networks, without worrying about hardware limitations (Vijh, 2020). Moreover, a number of stock exchanges make their APIs available, thus enabling real-time algo-trading. All of this has led to a great increase in recent research on the use of new machine learning models for stock market price prediction (Strader, 2020).

Initially, all kinds of econometric models such as ARIMA were used for this purpose (Ho, 2021). The development of technology has allowed increasingly complex machine learning models to be applied to time series prediction (Ho, 2021). Deep neural networks models are also increasingly used (Prasad, 2021). Their architecture is also evolving, which means that there are many different types of neural networks. Mostly recurrent neural network (RNN) models such as GRU and LSTM are used (Sonkiya, 2021).

Recently, a new architecture has emerged, which will be explored in the study. It is Generative Adversarial Network introduced by Goodfellow (2014). They were initially intended to generate synthetic images. However, later research has shown that they can be successfully used to generate future stock market valuations. Sonkiya (2021) used S-GAN with sentiment analysis of financial news as an input for the generator. The finBERT model, trained on financial data, was used for sentiment analysis. They managed to create a model that wins against traditional approaches such as ARIMA, LSTM, and GRU. Kumar (2021) used phase-space reconstruction (PSR) with GAN and compared it to LSTM with PSR as well. They have tested multiple time window sizes and epochs and managed to reduce the training time of the system compared to traditional approaches while maintaining satisfactory results.

Staffini (2022) proposed a DCGAN architecture with a CNN-BiLSTM discriminator and CNN generator. The models were compared with traditional approaches such as ARIMAX-SVR, Random Forest, and LSTM on 10 different stock market instruments. Both in single-step and multi-step prediction, the proposed DCGAN outperformed typical approaches. Zhang (2019) proposed GAN with LSTM as a generator and MLP as a discriminator. They have used pure financial data from yahoo finance as generator input. Then, the proposed model has been compared with ANN, SR, and LSTM. In this case, GAN was also found to be superior to the traditional approach.

Jiang (2021) has proposed a novel GAN architecture. It is called SF-GAN and consists of State Frequency Memory Neural Network (SFM) as a generator and CNN as a discriminator. He was able to achieve a significant improvement in the trend prediction of a financial instrument and minimize the prediction error. Lin (2021) has proposed two versions of GAN for time series. The first one was similar to Sonkiya's S-GAN with GRU as a generator and CNN as a discriminator. They have extended this idea and included WGAN-GP, a Wasserstein GAN with Gradient Penalty. Compared with GAN, WGAN-GP doesn't have a sigmoid function and it outputs a scalar value instead of probability.

Analyzing the results of the aforementioned research, it may be stated that GAN algorithms give very promising results. For example, Sonkiya's (2021) S-GAN model managed to get RMSE 62% smaller than LSTM, getting for S-GAN and LSTM RMSE equal to 1.827 and 2.939 respectively. Therefore we would like to apply GAN in predicting stocks. The main aim of the research is to build a model for predicting asset prices, based on GAN. Even though in previous papers the main measure for the performance of the models were forecast error measures, in the study, it was decided to include an investment strategy. This was done in order to prove the real application of the model. Based on a comparison of the prediction of the closing price for the next day and the closing price of today the proposed system makes a decision to sell, buy or hold the asset. The system is then evaluated and compared with Buy and Hold strategy.

In order to train GAN algorithm for predicting prices, two sources of data would be considered. First, downloaded stock market data from yahoo finance. Selected indicators are used to predict the explanatory variable, which in this case is the closing price of a given candlestick. This data is also used to calculate technical analysis indicators: simple moving average (SMA), exponential moving average (EMA), weighted moving average (WMA), Bollinger Bands (BB), and moving average convergence divergence (MACD).

Another source of information will be retrieved from sentiment analysis. Sentiment analysis is a method of gauging people's feelings about a company or its products, it should be considered important in the game industry, because the opinion of players expressed by posts on forums and the decision to buy a particular game may significantly affect the financial results of the company. Sentiment analysis will be performed on text data from the Reddit portal - one of the largest Internet forums of this kind. Reddit has subforums that allow aggregation of information by topic. This will allow to download comments on selected keywords from a site dedicated only to games. These keywords will be the names of companies and their most popular games. Google's BERT model will be used to analyze sentiment. It is a model pre-trained on millions of texts and considered as state-of-the-art model in the field of NLP (Devlin, 2019) (Sonkiya, 2021).

Afterwards, stock market data, technical analysis indicators, and sentiment analysis will be used to predict the one day ahead closing price of a given company. As for features, variables from the previous 30 days will be used. Based on the model's predictions, an investment strategy will be prepared. Buy, hold and sell decisions are deducted from the comparison of the next day's prediction and current price. We have taken into account transaction costs, the different cut-off points required for a trade, and the possibility of potential short selling. An α parameter has been included, ensuring that a buy/sell signal is generated when the price prediction of the next day and the previous day's price differs by α. The strategy is evaluated and compared to the Buy and Hold strategy.

In order to find the best solution, we have decided to verify which assumption should be confirmed. For this purpose, the following hypothesis has been stated:

Hypothesis 1 (H1) *Using data from sentiment analysis increase model predictions quality measured by MAE.*

Hypothesis 2 (H2) *The use of technical analysis indicators increase model predictions quality measured by MAE.*

One of the objectives of the paper is to prove that the GAN architecture is applicable to more than one company. This will show its potential generalisability and increase the range of practical applications. Therefore following hypotheses were stated:

Hypothesis 3 (H3) *The model architecture can be successfully applied to more than one company.*

Moreover, the investment strategy has been thoroughly examined. Our aim was to show that it can outperform the simplest market strategy - Buy & Hold. Moreover, different variants of the strategy have been tested, including those containing the possibility of short selling.

Hypothesis 4 (H4) *The investment strategy developed based on the GAN predictions is able to outperform the Buy&Hold strategy.*

Hypothesis 5 (H5) *Adding the possibility of short selling to the investment strategy increases the profitability of the system.*

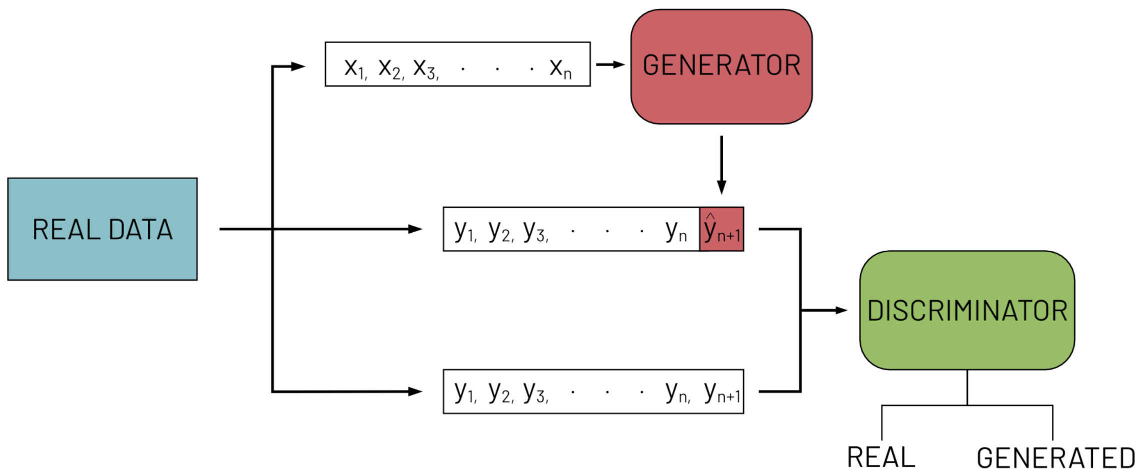
2 Proposed framework

A framework of the approach is shown in Figures 1 and 2. It consists of two main parts: a predictive model and an investment strategy. The model is based on a GAN algorithm consisting of two competing networks, a discriminator and a generator (Godfellow 2014). The generator tries to generate a prediction as close as possible to the actual closing price, and the discriminator tries to learn to recognize the data artificially generated by the generator from the actual data.

Three different sets of inputs used for the generator were examined: Stock market data only, Stock market data with technical analysis indicators, and Stock market data with technical analysis indicators and sentiment analysis results. The independent variables from the previous 30 days are used to predict the dependent variable which is the closing price. Once the model is trained, the generator itself is used for the prediction. This is due to the fact that the goal of GAN during training is to generate such predictions by the generator that the discriminator is unable to distinguish between real and generated data predictions from the generator.

The second part is the investment strategy. The prediction of the closing price for the next day is compared with the closing price of today. Based on their difference, the proposed system makes a decision to sell, buy or hold the asset. This chapter will present the detailed components of our framework.

Figure 1: GAN training architecture

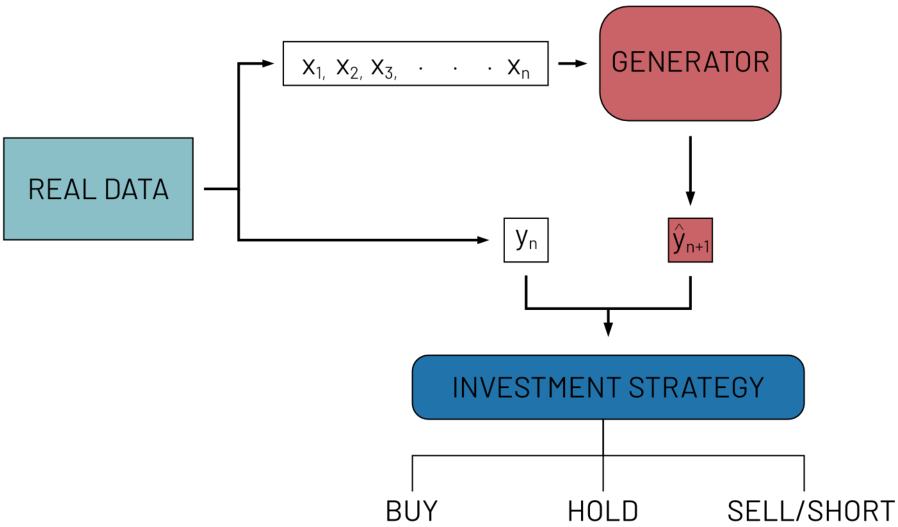


Source: Based on own research.

where:

X_n = independent variables vector on day n,
$\hat{y}_{n+1}$ = predicted close price in day n+1,
y_n = actual close price in day n,

Figure 2: Investment strategy architecture



Source: Based on own research.

where:

X_n = independent variables vector on day n,
$\hat{y}_{n+1}$ = predicted close price in day n+1,
y_n = actual close price in day n,

2.1 Generative adversarial network

Generative Adversarial networks are a family of neural networks first proposed by Godfellow (2014). The main field in which they are used is computer vision, but since their inception, various modifications of them have been tested in different fields, such as text-to-image translation and synthetic data generation (Wang, 2020) (Yilmaz, 2022). They are currently being tested in time series prediction as well (Zhang, 2019). GAN consists of two neural networks competing against each other in a zero-sum game. The generator tries to generate data as similar to the real data as possible, and the discriminator tries to recognize which data is real and which is generated. In other words, the generator tries to minimize the difference between the true and synthetic distributions, while the discriminator tries to maximize this difference. This can be represented by the following min-max function (Sonkiya, 2021):

GAN min-max function:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

In classical GAN models, the input to the generator (G) would be a latent vector derived from N(0, 1). In the proposed approach this is replaced by a vector consisting of the sentiment vector (Kumar, 2021), data from the stock market, and technical analysis indicators for early convergence (Lin, 2021). Its task is to generate a vector G(x) as similar as possible to the original distribution. In our system, due to its ability to deal with time series, the GRU network was chosen as the generator (Sonkiya, 2021).

2.2 Generator

Gated Recurrent Unit is an extended version of Recurrent Neural Networks which addresses the RNN vanishing gradient problem (Dey, 2021). Its architecture has been proposed by K.Cho in 2014 (Cho, 2014). Its name comes from gating mechanisms that allow the perceptron to choose which information should be saved or forgotten. It is similar to Long short-term memory networks, yet it lacks an output gate. This difference makes GRU less computationally expensive while maintaining similar or even better performance on smaller datasets.

Our proposed generator consists of three GRU layers containing 1024, 512, and 256 neurons, respectively. Each of them has a recurrent dropout of 0.2. These are followed by three multilayer perceptron (MLP) layers containing 128, 64, and 1 neuron, sequentially (Sonkiya, 2021) (Lin, 2021). The generator loss function is shown as follows:

Generator Loss Function:

$$-\frac{1}{m} \sum_{i=1}^{m} \log\left(D\left(G\left(x^i\right)\right)\right) \quad (2)$$

2.3 Discriminator

The discriminator in the model is a network composed of a 1-dimensional Convolutional Neural Network. Convolutional Neural Network is a class of artificial neural networks. It is widely used in many different fields, including computer vision, speech processing, and text processing (Alzubaidi, 2021). One of its main advantages is the ability to identify important features without previous indications. It takes its name from mathematical linear operations called convolution (Yamashita, 2018).

The network was chosen as the discriminator due to its differentiating capabilities. Its task is to discriminate between synthetic and real data. It assigns 1 to the values coming from the true distribution and 0 to the false one. The proposed discriminator consists of three convolutional layers. The first contains 32 units and has a kernel size of 3. The second contains 64 units and has a kernel size of 5. The third contains 128 units and has a kernel size of 5. Each has strides of 2 and a Leaky ReaLU activation function with an alpha parameter of 0.1. Then, there is a flatten layer that flattens the input. It is followed by three multilayer perceptron (MLP) layers containing 220, 220, and 1 units, sequentially (Sonkiya, 2021) (Lin, 2021). The discriminator loss function is shown as follows:

Discriminator Loss Function:

$$-\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(y^{i}\right)+\log\left(1-D\left(G\left(x^{i}\right)\right)\right)\right] \tag{3}$$

The optimizer used for both Generator and Discriminator was ADAM and the learning rate was set to 0.0016 (Sonkiya, 2021).

Due to the use of neural network based architectures in the study, rescaling of the data was required. For this purpose, min-max normalization was used. The equation for the rescaled value is:

$$x_{\text{scaled}}=\frac{x-min(x)}{max(x)-min(x)} \tag{4}$$

To avoid overfitting and evaluate models and investment strategies on data that was not used during training, it was decided to split our dataset into train dataset and test dataset with a 75%/25% ratio. A training set was used to train the GAN model and to choose the best parameters for investment strategy. Then, a test dataset was used to check findings on data that hasn't been seen by the system before.

2.4 Evaluation of the model

Evaluation of the model is going to be done using two metrics, Root Mean Square Error and Mean Absolute Error, which may be defined as follows:

Root Mean Square Error (RMSE):

$$RMSE=\sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i-x_i)^2} \tag{5}$$

where:

x_i = true value
y_i = predicted value
n = number of observation

Mean Absolute Error (MAE):

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - x_i| \tag{6}$$

where:

x_i = true value
y_i = predicted value
n = number of observation

2.5 Investment Strategy

An investment strategy based on the model predictions was also developed in the study. This strategy makes several important assumptions:

- on a given day t the system is able to buy an asset at close price at the very end of the day,

- it is possible to enter a short trade on the asset,

- at the start of the experiment 1000 dollars are available for investing,

- all cash resources are used for each trade,

- trade fee is equal to 0.0007$, [1]

In the strategy, buy and sell signals are based on a comparison of a real today's price and a prediction of the price on the next day. The basic approach is to buy when $\hat{y}_{t+1} > y$ and sell when $\hat{y}_{t+1} < y$. However, this approach has a major drawback as it involves a very large number of transactions, which, taking into account transaction fees, leads to high costs. For this reason, the strategy includes an α parameter which ensures that a buy/sell signal is generated when the price prediction of the next day and the previous day's price differs by α. As a result, trades are executed less frequently and potentially lead to a more profitable solution. It prevents funds from being consumed by transaction fees. In the strategy different α parameters may be considered for the buy and sell signals. The situation in which the possibility of short transactions is used is also tested. In this case, a short trade is entered along with a sell signal. Different versions of α parameters for buy and sell signals were tested and then the best one was selected for each company.

The diffrence of prices was calculated using following formula:

$$diff_{t+1} = \hat{y}_{t+1} - y_{t1} \tag{7}$$

where:

[1] Transaction fee comes from NASDAQ transaction fee in IterativeBrokers (https://www.interactivebrokers.com/en/index.php?f=948)

$diff_{t+1}$ = calculated diffrence,
$\hat{y}_{t+1}$ = predicted price of asset in next day,
y_t = actual price of asset in current day,

The buy/selll/hold signal are made as following:

$$signal = \begin{cases} buy & diff_{t+1} > y_t \cdot \alpha_{buy} \div 100 \\ sell & diff_{t+1} < -(y_t \cdot \alpha_{sell} \div 100) \\ hold & otherwise \end{cases} \tag{8}$$

where:

$diff_{t+1}$ = calculated diffrence
α_{buy} = alpha parameter for buy signal
α_{sell} = alpha parameter for sell signal

2.6 System evaluation

To evaluate the proposed system, it is analyzed at the end of the test period. The chosen metric is calculated as the sum of held assets multiplied by their last price and amount of cash at the end of the test time period. The reason for this is that the system only allows full shares to be purchased. Therefore, it was possible to have the change from the transaction in cash. It is compared with the initial budget and the result of the Buy and Hold strategy. This allows to see if the developed system performs better than holding cash or buying an asset and leaving it for the whole period of time.

3 Data

3.1 Stock Data

In the study it was decided to train the algorithm on stocks of 4 large companies from the gaming industry: Electronic Arts, Ubisoft, Take-Two Interactive Software, and Activision Blizzard, called in the later part of the paper by their stock tickers: EA, UBSFY, TTWO, ATVI. This gaming sector was chosen due to the fact that the study investigates the influence of sentiment analysis - one of our hypotheses is that the opinions of Reddit users have a particularly significant impact on the valuations of a given company in the gaming industry. The selection of more than one company was intended to show that the algorithm is reproducible and universal. These companies are among the leaders in the gaming industry. The choice of only U.S. companies was caused by a possible language barrier when analyzing the sentiment of companies that produce games primarily for the Asian market. Due to the use of one-day intervals, limitations of Reddit's API (Application Programming Interface), and limitations of computing power, the analysis was conducted only over a nearly three-year period: from 01/01/2019 to 31/10/2021. Data is analyzed in 1 day periods. The data comes from the US NASDAQ stock market and was retrieved using the yahoo finance library for the python language. In Table 1. descriptive statistics for each of the previously mentioned companies have been presented.

Table 1: Descriptive statistics for chosen companies

	Company	ATVI	EA	TTWO	UBSFY
Open	count	178.0000	178.0000	178.0000	178.0000
	mean	83.8875	139.3584	170.9768	12.8549
	std	11.4400	5.3196	10.6307	1.8079
	min	56.9778	120.4896	145.9100	9.0500
	25%	77.6000	137.5287	163.1550	11.1675
	50%	83.5100	140.7925	171.3150	12.9601
	75%	93.5850	142.8999	178.8850	14.2100
	max	98.8600	148.5431	192.2600	16.0996
High	count	178.0000	178.0000	178.0000	178.0000
	mean	84.7118	140.8630	172.9497	12.9261
	std	11.4490	5.2717	10.6649	1.8045
	min	57.4400	123.5934	147.1000	9.2700
	25%	78.3725	139.5444	164.6825	11.2589
	50%	84.4625	142.1893	173.3464	13.1050
	75%	94.7925	144.0306	182.2375	14.2800
	max	99.4590	148.5531	195.8250	16.1800
Low	count	178.0000	178.0000	178.0000	178.0000
	mean	82.7644	137.7310	168.8698	12.7556
	std	11.4891	5.4720	10.3637	1.8032
	min	56.4000	119.9184	144.5810	9.0500
	25%	76.6675	136.2402	160.5650	11.0476
	50%	82.6550	139.2077	169.4100	12.8400
	75%	92.7575	141.2427	176.5900	14.1275
	max	97.6100	145.7801	186.4200	16.0500
Close	count	178.0000	178.0000	178.0000	178.0000
	mean	83.6483	139.1895	170.8772	12.8366
	std	11.4663	5.4726	10.5099	1.8054
	min	57.2800	120.0682	145.2500	9.1400
	25%	77.3650	137.8033	162.8950	11.1625
	50%	83.4700	140.6691	171.2800	12.9750
	75%	93.4025	142.6030	178.9275	14.2174
	max	99.1800	148.1741	192.9100	16.1300
Volume	count	178.0000	178.0000	178.0000	178.0000
	mean	8236266.1966	2473996.9831	1276613.7528	169367.5056
	std	5413813.4429	1073436.8484	669500.1064	278989.7716
	min	2596873.0000	1016396.0000	559480.0000	16495.0000
	25%	5105944.5000	1766566.2500	876545.7500	46863.2500
	50%	6667305.0000	2181122.5000	1095232.0000	69654.5000
	75%	9477788.5000	2854991.5000	1422011.2500	148630.0000
	max	43753567.0000	8344344.0000	5935523.0000	1833827.0000

Source: Based on own research and yahoo finance stock data.

3.2 Text Data

As it was mentioned before sentiment analysis is part of the research. As in the study, only the companies from the gaming industry are considered, it has been decided to use data coming from the reddit.com portal, which is one of the largest forums in the world. It also has a feature that helps to obtain data for the study - it is divided into parts, the so-called subreddits, which gather people interested in a particular topic. In the study, the r/Games subreddit is chosen as the main source of text data. This forum has 3.1 million users and millions of comments about games.

Next, in order to perform sentiment analysis, for each company selected, a given set of keywords was chosen to retrieve the data. The keywords were names of the most popular game series of a particular publisher and the publisher's name itself. This allows us to analyze the opinions of users about games created by a given company. Next, using python's psaw library, all the comments that had been posted over a predefined period of time on the r/Games subreddit containing the keywords mentioned below were gathered. In Table 2. chosen keywords for each considered company have been presented.

Table 2: Key words chosen for each considered company

	EA	TTWO	UBSFY	ATVI
0	EA	Take Two	Ubisoft	Blizzard
1	Fifa	NBA 2K	Assasin's Creed	Starcraft
2	The Sims	Battleborn	AC	Warcraft
3	Need for Speed	BioShock	Far Cry	Overwatch
4	NFL	Borderlands	Watch Dogs	Diablo
5	Apex	Evolve	Rainbow Six Siege	World of Warcraft
6	Battlefield	Mafia	Wildlands	Hearthstone
7	Bejeweled	Civilization	For Honor	Heroes of the Storm
8	Battlefront	The Darkness	Tom Clancy's	
9	NBA	XCOM	The Division	
10	Dragon Age	WWE		
11	Titanfall	GTA		
12	Dead Space	Grand Theft Auto		
13		Max Payne		
14		Red Dead Redemption		
15		RDR		

Source: Based on own research.

3.3 Sentiment Analysis

Sentiment analysis is a process of extracting users' feelings and emotions (Liu, 2010). It is a part of Natural Language Processing. It is designed to determine with a given probability whether a given statement was positive, negative, or neutral. Then such predictions are converted into numerical data, returning [1, 0) for positive sentiment, 0 for neutral, and (0, -1] for negative. There are many different types of models for sentiment analysis. In our study, the BERT (Bi-Directional Encoder Representations from Transformers) model created by Google researchers was used (Devlin, 2019).

BERT is a state-of-the-art NLP model. One of its biggest advantages is taking whole sentences as an input in contrast to traditional NLP models that take one word at a time. For this reason, it is referred to as Bi-Directional. In this way, it can also learn the context between the words in a sentence. One of BERT's biggest advantages is that it is a semi-supervised model. It is pre-trained on very large sets of non-labeled data, learning to fill gaps in the text. In this way, it is forced to identify the masked word based on the context. It also uses an attention mechanism that allows it to learn the relationship between words (Devlin, 2019). Then this model can be trained for any task just by adding one extra layer. All this makes BERT an extremely versatile model and very well suited to sentiment analysis without the need for large computational resources.

Due to the occurrence of many comments concerning a single company on a given day, it was necessary to group them. All comments containing a keyword related to a given company were combined into one matrix. For each of the comments, sentiment has been assigned. In the next step, sentiments were aggregated by day of posting into the following vector for each day:

$$\text{Sentiment vector for day } i\text{:}$$
$$[n, \mu, \sigma, med, Q_1, Q_3] \tag{9}$$

where:

n = number of comments related to chosen company on day i
med = median of all sentiment values related to chosen company on day i
μ = mean of all sentiment values related to chosen company on day i
σ = standard deviation of all sentiment values related to chosen company on day i
Q_1 = first quantile of all sentiment values related to chosen company on day i
Q_3 = third quantile of all sentiment values related to chosen company on day i

3.4 Technical Analysis

Technical analysis (TA) is the most commonly used method in automated trading (Stanković, 2015). Its indicators describe the behavior of a given stock market instrument based on mathematically processed recent observations (Stanković, 2015). It is used in the analysis for several reasons. The first is the ability to obtain early convergence, thus reducing the time to train the model and the resources needed to do so. Moreover, such wide use of these indicators among other stock exchange participants allows the system to take into account possible behavior of the price caused by the results of technical indicators. In the study, the indicators described below were taken into account:

- Moving Averages - Several types of moving averages are used. The first and simplest of them is a simple moving average (SMA) (Wong, 2003). It takes into account equally weighted observations in a given time window. For obvious reasons, one may assume that closer dates may have a more significant influence on future prices. Therefore, in addition to the SMA, Weighted Moving Average (WMA) (Perry, 2010) and Exponential Moving Average (EMA) (Grebenkov, 2014) were used. The WMA solves the aforementioned problem by giving more weight to more recent data. EMA works in a similar way, but the price change is not consistent but exponential.

- Bollinger Bands - Bollinger Bands consist of three bands (Lento, 2007). The middle one is a moving average. Higher and lower bands are deviated from the middle one by 2 standard deviations up and down respectively. Each of these three bands is used as an input in the GAN model.

- Moving Average Convergence Divergence - Moving Average Convergence Divergence (MACD) consists of two lines (Chong, 2008). The core of the indicator is the MACD line which is the difference between the 12-period EMA and the 26-period EMA. The second line is the signal line which is a 9-period EMA. Their position relative to each other helps to determine whether the market is oversold or overbought.

In the study, two types of data were used - Stock data and text data. Technical analysis indicators were calculated on the basis of the Stock data. Sentiment analysis has been conducted on text data. In this way three sets of data were created - Stock market data only, Stock market data with technical analysis indicators, and Stock market data with technical analysis indicators and sentiment analysis results. It will allow the quality of the models to be tested on different datasets and the optimal one to be selected. The results will be presented in the following chapter.

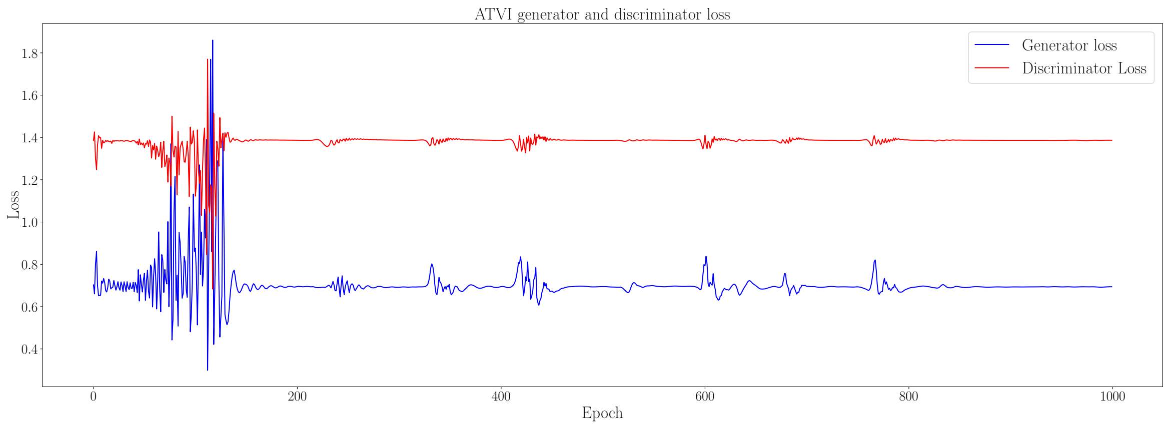
4 Results of framework

4.1 GAN results

The model was trained using the Python language frameworks for NN modeling - TensorFlow (Abadi, 2016) and Keras (Chollet, 2015) (Gulli, 2017). The hardware used for training consisted of an Nvidia GeForce GTX 1070 graphics card, an Intel i5-6600 processor, and 16 GB of RAM. The training was performed together with the use of CUDA cores. Each training was run with an epoch parameter of 1000. Hyper-parameter optimization was also performed, however, due to the limitation of computing power, it could not be extensive. It was limited to checking the behavior of the network after removing the MLP layers from both the generator and the discriminator. This did not give the desired results, therefore in the further part of the study, the parameters proposed by Sonkiya (2021) were used. The behavior of the model with different types of input data for the generator was also checked. Three datasets were used as an input for the generator - Stock market data only, Stock market data with technical analysis indicators, and Stock market data with technical analysis indicators and sentiment analysis results.

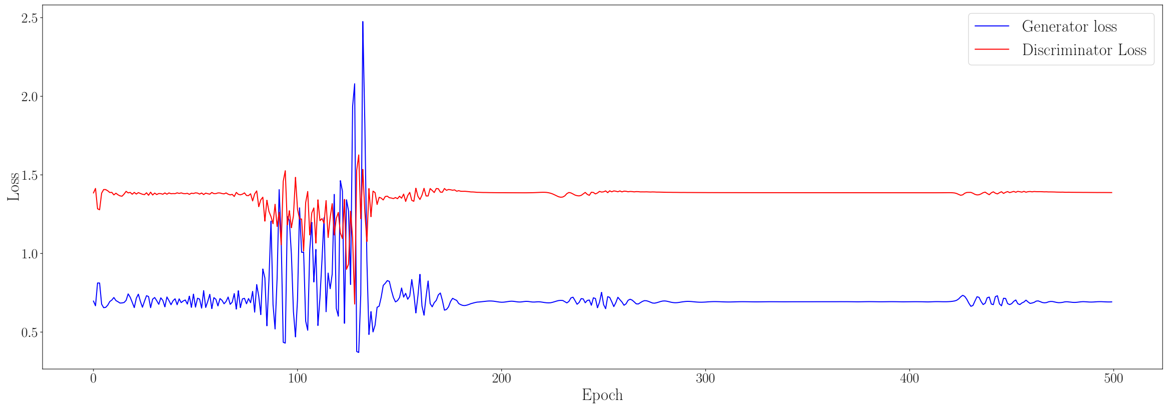
To analyze the process of training, the loss function of the best-performing model for every company was inspected. In this case, all of them appeared to be trained on Stock data with technical analysis indicators. It is shown in Figures 7-10. They represent the change in the value of the generator and discriminator loss functions over time. It is easy to see how they compete with each other. It can also be seen that changes in the loss functions occur simultaneously, i.e. if the generator loss changes then the discriminator loss also changes. In any case, around epochs 100-300 there is a period of strong instability, characterized by very strong changes in both loss functions. It usually lasts several tens of epochs and ends with stabilization of both functions. In 3 out of 4 cases the period of severe instability occurs only once, the exception is UBSFY where such periods occur twice. After the initial periods of instability, the loss function becomes stable, only experiencing subtle changes once in a while. Once the loss function has stabilized, such a model is ready to be used in an investment strategy. It follows that with our proposed framework it is necessary to train the model for at least 400 epochs. Once the models were trained, they were evaluated on test sets.

Figure 3: ATVI generator and discriminator loss



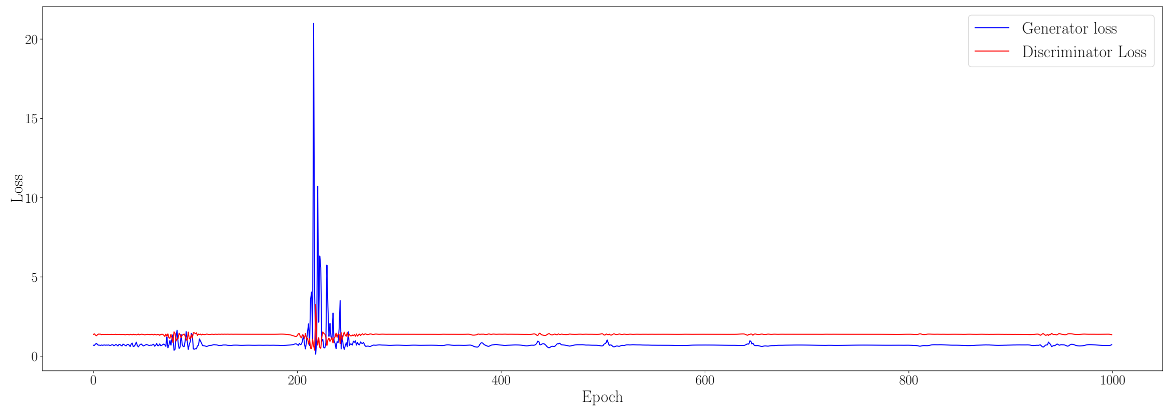
Source: Based on own research.

Figure 4: EA generator and discriminator loss



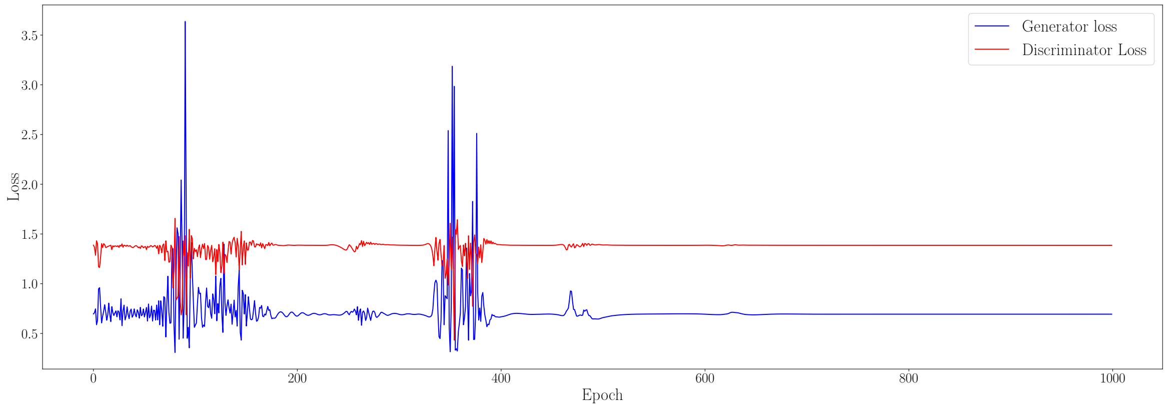
Source: Based on own research.

Figure 5: TTWO generator and discriminator loss



Source: Based on own research.

Figure 6: UBSFY generator and discriminator loss



Source: Based on own research.

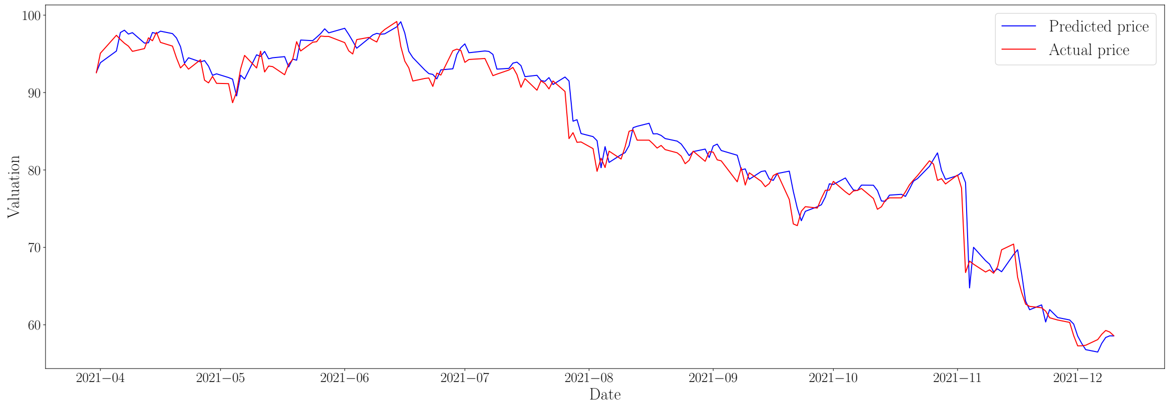
In Table 3. the results of forecast error metrics have been presented alongside the mean close price of a given company in the test dataset. By far the best model was found to be the one without sentiment and with technical analysis. One reason why the use of sentiment may not be correct is the noise and data impurity from Reddit. However, this topic needs further exploration with more computational resources. The forecast errors are shown together with the average share price of a given company in the test dataset. The mean has been included here due to the relativity of error. For each of the companies, the MAE does not exceed 2.1 % of the average price value over the test period when using Stock data alongside technical analysis indicators. The best performer here is the model for EA, for which this value is 0.95%. In the case of this company, the model performed best when the epoch parameter was changed from 1000 to 500. Next, a graphical analysis of the results has been carried out for the best-performing model for each of the companies.

Table 3: Forecast error metrics for chosen companies

Dataset	Metric	ATVI	EA	TTWO	UBSFY
Stock data	MAE	7.36249	7.80082	8.96674	1.31115
	RMSE	9.53919	8.33803	10.83445	1.67089
Stock data with TA	MAE	1.35850	1.34270	3.54050	0.25780
	RMSE	1.88670	1.89560	4.50690	0.33760
Stock data with TA	MAE	7.69750	3.17484	3.67834	2.67735
and sentiment	RMSE	7.95309	4.14371	4.66893	3.12417
Close Price Mean		83.64834	139.18953	170.87719	12.83657

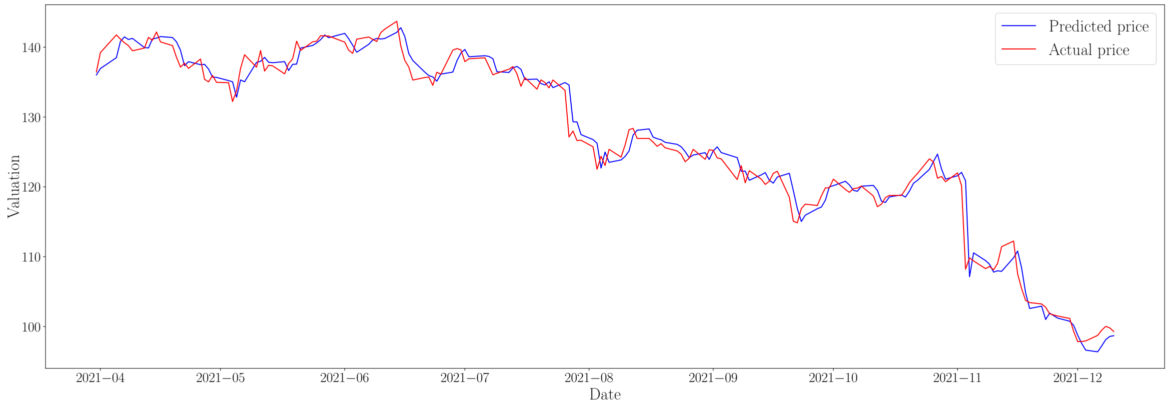
A comparison of predicted and actual prices is shown in Figures 3-6. An important fact is the crossing of the prediction and actual price lines on each of the graphs showing the differences between the predicted value and actual value. This is crucial because if these lines did not intersect, it would mean that the model constantly predicts too high or too low value. In Figures 3, 4, and 6 it can be seen that 3 of the 4 models perform well in forecasting future prices. The three charts are similar, the *predicted price* and *actual price* lines are very close to each other and often intersect. However, forecasting TTWO prices shown in Figure 5 proved problematic. It can be seen here that the *predicted price* line deviates significantly from the *actual price*. Nevertheless, the model performs acceptably even here. The graphical analysis is confirmed by the ratio of MAE to average price, which in the case of TTWO looks the worst, but does not exceed the aforementioned 2.1 %.

Figure 7: ATVI actual price vs predicted price



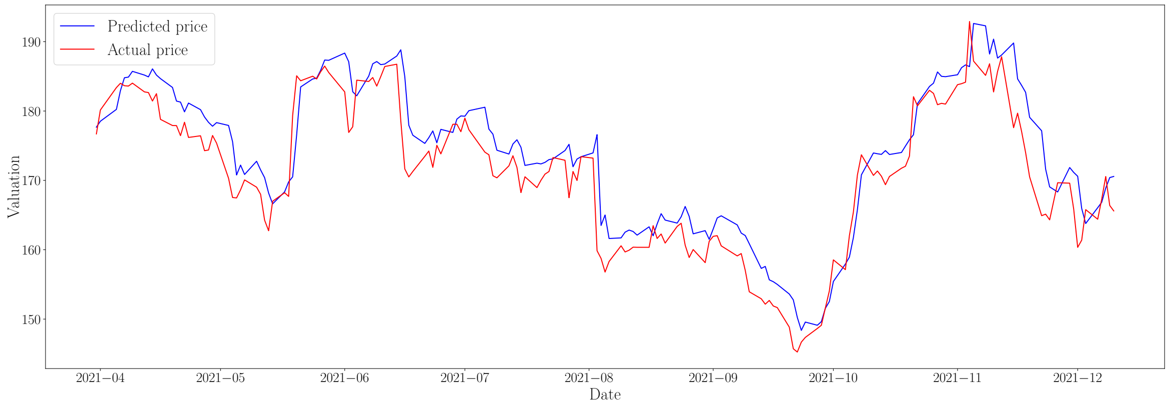
Source: Based on own research and yahoo finance stock data.

Figure 8: EA actual price vs predicted price



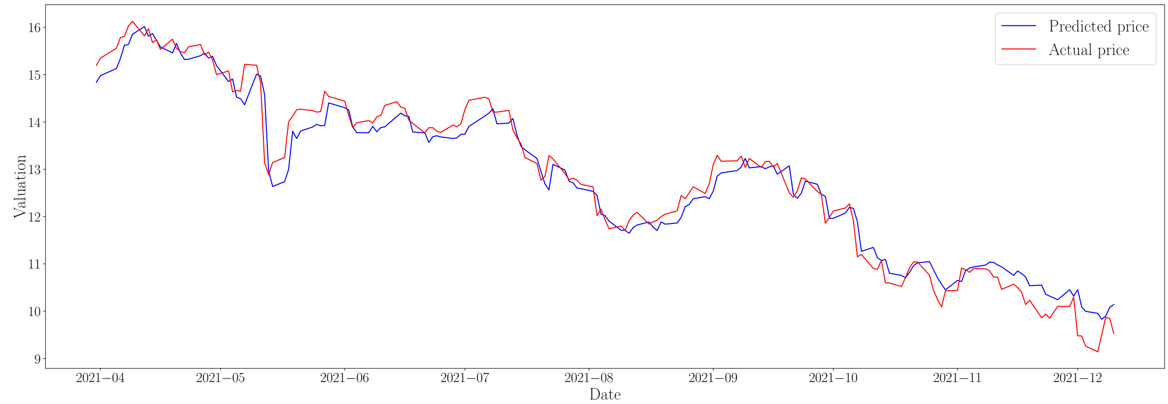
Source: Based on own research and yahoo finance stock data.

Figure 9: TTWO actual price vs predicted price



Source: Based on own research and yahoo finance stock data.

Figure 10: UBSFY actual price vs predicted price



Source: Based on own research and yahoo finance stock data.

4.2 Investment strategy results

Once satisfactory models were created, the results of our investment strategy were analyzed. Parameters α_{sell}, α_{buy} which decide the required percentage of price change when selling and buying, and *if_short* deciding whether the short selling is used were evaluated. The aim was to find parameters that maximize the return on investment.

Table 4: Signal hyper parameters

α_{buy}	α_{sell}	if_short
0.0	0.0	True
0.2	0.2	False
0.4	0.4	
0.6	0.6	
0.8	0.8	
1.0	1.0	
1.2	1.2	
1.4	1.4	
1.6	1.6	
1.8	1.8	
2.0	2.0	
2.2	2.2	
2.4	2.4	
2.6	2.6	
2.8	2.8	
3.0	3.0	
3.2	3.2	
3.4	3.4	
3.6	3.6	
3.8	3.8	

To select the best possible set of parameters, hyperparameter optimization was carried out. In order to avoid matching the strategy with the test results, this was performed on the training data. Using the parameters shown in Table 4, all their possible permutations were created. The algorithm for Backtesting was then run with each possible permutation and the set of parameters maximizing the balance at the end of the training period was found. The selected parameters for each company are presented in Table 5.

Table 5: Best strategy hyper parameters

ticker	α_{buy}	α_{sell}	if_short
UBSFY	1.8	1.4	True
EA	2.6	0.6	True
TTWO	1.6	0.4	True
ATVI	0.0	2.2	True

The parameters for the investment strategy found in this way are subject to some bias. It is likely that the predictions of the test set would be less precise than the train set. Therefore, an additional set of parameters was arbitrarily chosen. The parameter

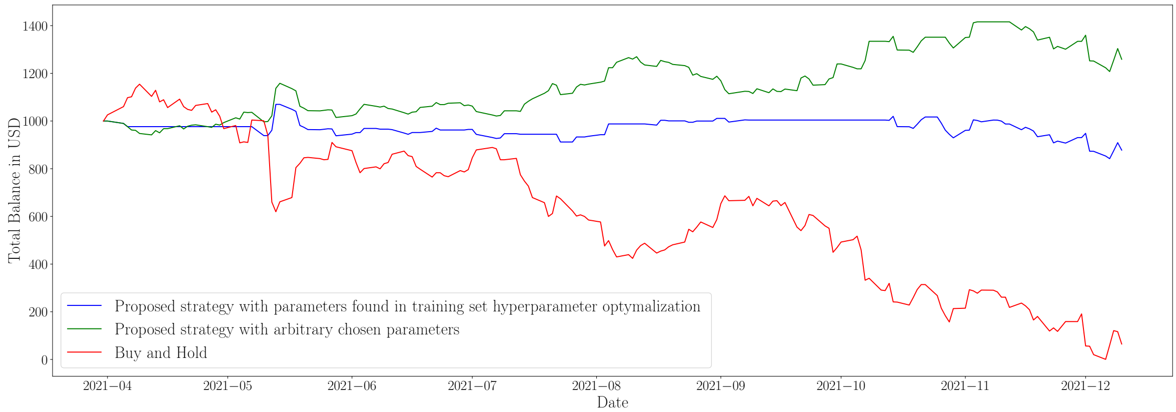
α_{buy} was set up to 2 and α_{sell} to 0. A short selling option has also been made available. A high top-cut-off was chosen with a conservative approach for buying, making trades less frequent and therefore funds are not used up by transaction fees. A zero down-cut-off will instead give an instant sell if the model prediction is smaller than the previous price. This is intended to minimize losses. Since these parameters are chosen purely on the basis of the researcher's judgment they may require deeper analysis. One idea that could be further investigated is to split data into one more dataset that would be used exclusively to select parameters for a trading strategy that would be tested on a real test set that the system has not seen.

Table 6: Diffrent strategies balance at the end of test time peroid

ticker	Buy and Hold	Proposed strategy with parameters found in hyperparameter optimization	Proposed strategy with arbitrarily chosen parameters
UBSFY	631.839 $	878.008 $	1258.925 $
EA	935.412 $	1006.768 $	1079.715 $
TTWO	944.500 $	876.744 $	870.907 $
ATVI	660.481 $	631.715 $	1110.533 $

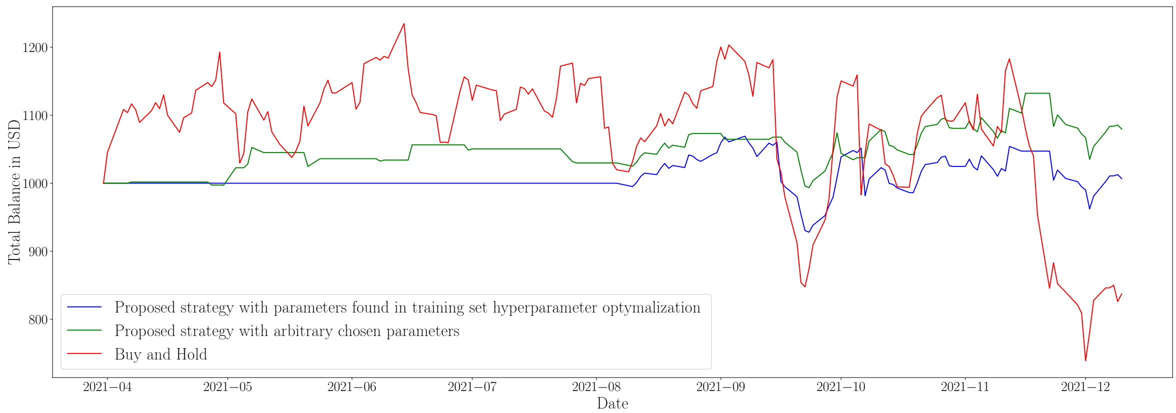
As can be seen in Figures 11-14, in 3 out of 4 cases the presented strategy outperformed the Buy and Hold strategy. Interestingly, the arbitrarily chosen parameters for the investment strategy turned out to be better than those found by hyperparameter optimization. However, this agrees with our assumption that due to differences in model prediction on the training and test sets, these parameters may not be optimal on data that our system has not seen. This is clearly visible in the case of UBSFY and ATVI, where in both cases strategy with arbitrarily chosen parameters outperforms strategy with parameters hyperparameter optimization. In the case of UBSFY, the percentage difference of the final balance between the strategies is 43% and in the case of ATVI - 76%. In the other two companies, EA and TTW, strategies with both sets of parameters gave similar results. The best results were achieved for UBSFY. In this case, the strategy with arbitrarily chosen parameters outperformed the Buy and Hold strategy significantly, reaching 1258$ at the end of the test period while the Buy and Hold only reached 631$. The worst results were obtained for TTWO. However, this is understandable, because already at the level of the model itself it was clear that MAE compared to average price values is the highest there. This means that this model is not good enough and needs further research. It is important to note that each of these companies was in a downtrend. For each company, the Buy and Hold strategy proved unprofitable. In the case of UBSFY and ATVI, the Buy and Hold strategy would lose more than 30% of initial funds. This makes it all the more optimistic that 3 of the 4 assets made significant gains.

Figure 11: UBSFY proposed strategy with diffrent parameters vs Buy and Hold



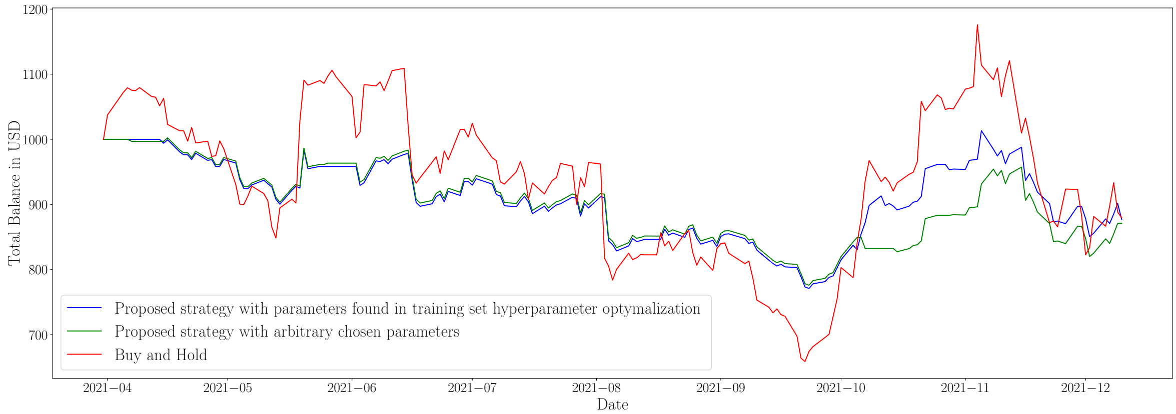
Source: Based on own research and yahoo finance stock data.

Figure 12: EA proposed strategy with diffrent parameters vs Buy and Hold



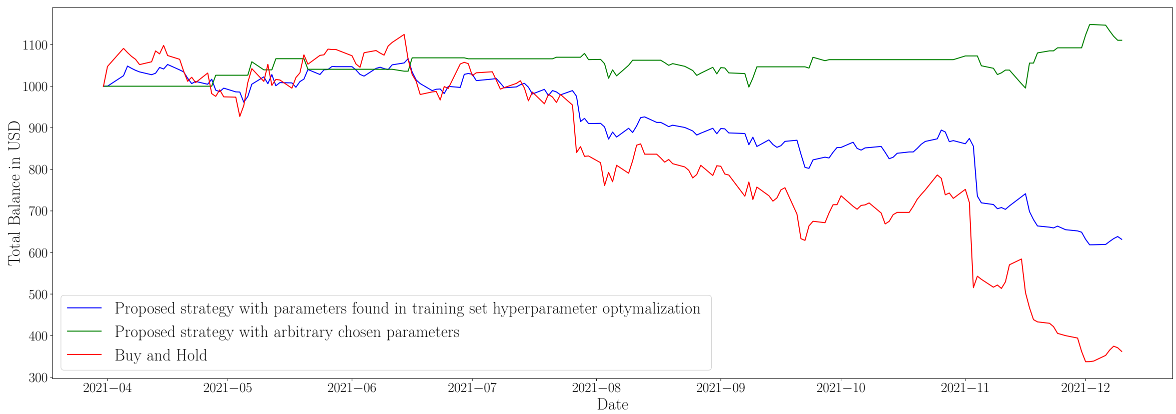
Source: Based on own research and yahoo finance stock data.

Figure 13: TTWO proposed strategy with diffrent parameters vs Buy and Hold



Source: Based on own research and yahoo finance stock data.

Figure 14: ATVI proposed strategy with diffrent parameters vs Buy and Hold



Source: Based on own research and yahoo finance stock data.

5 Conclusion

The aim of the study was to test the feasibility of using the GAN network and the BERT model to predict the prices of four companies from the game industry - Electronic Arts, Ubisoft, Take-Two Interactive Software, and Activision Blizzard. The BERT model was used to analyze the sentiment of comments about a given company and its games from reddit.com. These data along with technical analysis indicators and stock market data were then used as explanatory variables for the GAN model. GAN models without sentiment analysis and/or technical analysis indicators were also tested. Then, using the predictions from the model, an investment strategy was created. Optimization of the hyperparameters in this system was carried out in order to find profit-optimizing parameters. Stock data with technical analysis indicators turned out to be the best performing dataset. Using it, for each of the companies, the MAE does not exceed 2.1 % of the average price value over the test period. For 3 of the 4 companies, it was possible to create a strategy that significantly outperforms the buy and hold approach.

Additionally, the study tested the hypotheses raised in the introduction. The first hypothesis *(H1) Using data from sentiment analysis increase model predictions quality measured by MAE* has to be rejected since we haven't managed to create a well-performing model without throwing sentiment data away. Their use made the model perform much worse. This may have been due to flaws in the data retrieval API and noise in the text data. Another explanation may be that the valuation of these companies is not dependent on the opinion of the players.

The second hypothesis *(H2) The use of technical analysis indicators increase model predictions quality measured by MAE*, holds due to the fact that technical analysis improved each of our models. This matches our expectations that these indicators can provide the model with additional information and accelerate its training. Moreover, this may be influenced by the fact that a large number of stock market participants use Technical Analysis.

The third hypothesis *(H3) The model architecture can be successfully applied to more than one company*, holds due to the fact that we were able to apply it to 3 out of 4 tested companies with good results. The only company where the application of our model has been only partially successful is TTWO. Nevertheless, it was shown that the model can be generalized and that it is possible to apply model architecture to more than one company.

The fourth hypothesis *(H4) The investment strategy developed based on the GAN predictions is able to outperform the Buy&Hold strategy* holds for 3 out of 4 companies that we have tested. The reason for this is that the TTWO model was surely the worst one and it needs further research. Since we have shown that the system can outperform the buy and hold approach, H3 cannot be rejected.

Finally, the fifth hypothesis *(H5) Adding the possibility of short selling to the investment strategy increases the profitability of the system*, holds due to the fact that strategies with the ability to short selling performed better. This has been influenced by the general downward trend valuation of companies during the test period, during which it is very effective to earn on short trades.

In conclusion, the main finding is the confirmation of the possibility to use GAN networks to create an effective system that allows us to outperform the Buy and Hold strategy and make a profit despite the fall in the price of an asset. The study has also shown that these networks have their challenges and there is a great scope for further research. The main ones are their instability during training and the unclearness of when to stop training. Better results could also be achieved with a more sophisticated investment strategy. Nevertheless, the results are promising and show that this system could have potential business applications.

6 Intentional self-criticism

During the study, certain problems were encountered. In the future, these should be investigated in more detail and checked whether their effect can be reduced.

The first of the problems posed by the existence of two loss functions is the difficulty of determining when a model is trained. Due to the fact that the two networks compete with each other, it is impossible to say unambiguously what values of the two loss functions characterize the best model. This is also a considerable difficulty in optimizing hyperparameters, as it is difficult to determine at which point training should be stopped.

Another problem is the different results with the same parameters. Neural networks are by definition not deterministic, as stochastic gradient descent and random assignment of initial weights can affect random network behavior and different results with the same parameters. They become deterministic after they have been trained and all weights are fixed. This effect is further compounded by the fact of using two networks competing with each other and having two separate loss functions. Further analysis can focus on stabilizing our system so that in business use it is possible to retrain it on new data without the risk of breaking the whole model.

The investment strategy also needs further analysis. In our study, a very simple method was used, which involves trading the entire funds on each trade. In order to minimize the risk of loss in the event of a sudden fluctuation, the system could be changed in the future to dose the amount of funds allocated to buy or sell transactions depending on the model's confidence in the price change.

References

[1] A. Prasad, A. Seetharaman, Importance of Machine Learning in Making Investment Decision in Stock Market, 2021

[2] M. Vijh et al, Stock Closing Price Prediction using Machine Learning Techniques, 2020

[3] T. Strader, Machine Learning Stock Market Prediction Studies: Review and Research Directions, 2020

[4] M K Ho, H. Darman, S. Musa, Stock Price Prediction Using ARIMA, Neural Network and LSTM Models, 2021

[5] A. Tealab, Time series forecasting using artificial neural networks methodologies: A systematic review, 2018

[6] J. Devlin et al, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019

[7] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 2014

[8] P. Dey et al, Comparative Analysis of Recurrent Neural Networks in Stock Price Prediction for Different Frequency Domains, 2021

[9] K. Cho et al, On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, 2014

[10] L. Alzubaidi et al, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, 2021

[11] R. Yamashita, Convolutional neural networks: an overview and application in radiology, 2018, 612

[12] J. Goodfellow et al, Generative Adversarial Nets, 2014

[13] J. Yoon, D. Jarret, M. Schaar, Time-series Generative Adversarial Networks, 2019

[14] P. Sonkiya, V. Bajpai, A. Bansal, Stock price prediction using BERT and GAN, 2021

[15] Z. Wang et al, Text to Image Synthesis With Bidirectional Generative Adversarial Network, 2020

[16] B. Yilmaz, Synthetic demand data generation for individual electricity consumers : Generative Adversarial Networks (GANs), 2022

[17] A. Kumar et al, Generative Adversarial Network (GAN) and Enhanced Root Mean Square Error (ERMSE): Deep Learning for Stock Price Movement Prediction, 2021

[18] H. Lin et al, Stock price prediction using Generative Adversarial Networks, 2021

[19] J. Stanković, Investment Strategy Optimization Using Technical Analysis and Predictive Modeling in Emerging Markets, 2015

[20] J. Michańków, P. Sakowski, R. Ślepaczuk, LSTM in Algorithmic Investment Strategies on BTC and S&P500 Index, 2022

[21] A. Staffini, Stock Price Forecasting by a Deep Convolutional Generative Adversarial Network, 2022

[22] Zhang et al, Stock Market Prediction Based on Generative Adversarial Network, 2019

[23] J. Jiang, Stock Market Prediction Based on SF-GAN Network, 2021

[24] H. Lin et al, Stock price prediction using Generative Adversarial Networks, 2021

[25] B Huang et al, Automated trading systems statistical and machine learning methods and hardware implementation: a survey

[26] Wing-Keung Wong, M. Manzur, Boon-Kiat Chew, How rewarding is technical analysis? Evidence from Singapore stock market 2003

[27] M. Perry, The Weighted Moving Average Technique, 2010

[28] S. Grebenkov, J. Serror, Following a trend with an exponential moving average: Analytical results for a Gaussian model, 2014

[29] C. Lento, N. Gradojevic, Investment information content in Bollinger Bands?, 2007,

[30] Terence Tai-Leung Chong, Wing-Kam Ng, Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30, 2008

[31] Bing Liu et al, Sentiment analysis and subjectivity, 2010

[32] M. Abadi et al, Tensorflow: A system for large-scale machine learning, 2016

[33] F Chollet et al, Keras, 2015

[34] A. Gulli, Deep learning with Keras, 2017