Game industry companies stock price prediction using GAN and BERT

Jakub Wujec

May 8, 2022

Abstract

In this article we propose the use of the GAN model to predict stock market behavior and an investment strategy using the results of the model. We will be helped by Technical Analysis and Sentiment Analysis serving as explanatory variables for our model. Our study was conducted on four companies in the game industry. This industry was chosen by us because of the potential impact of a company's customer reviews on its valuation. We investigate this by analyzing the sentiment of selected comments containing pre-prepared keywords. The sentiment is then calculated using the NLP model created by Google - BERT. GAN then tries to predict the future valuation of a given company using the variables mentioned above. This valuation is then used to provide buy or sell signals.

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, to make money. A not insignificant influence on this has been the facilitation of access to computing power, allowing the seamless use of algorithms such as neural networks, without worrying about hardware limitations. Moreover, a great number of 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 models for stock market price prediction.

In our study, we examine the capabilities of the Generative Adversarial Networks introduced in 2014 by J. Goodfellow [6]. 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. We use yahoo finance to download stock market data. From these we then select the indicators used to predict the explanatory variable, which in our case is the closing price of a given candlestick. This data is also used to calculate technical analysis indicators. These are simple moving average (SMA), exponential moving average (EMA), weighted moving average (WMA), bollinger bands (BB) and moving average convergence divergence (MACD).

Sentiment is one way of gauging people's feelings about a company or its products. It is even more important in the game industry, which we analyse in our work, because the opinion of players expressed by posts on forums and the decision to buy a particular game very significantly affects the finance results of the company. To download text data, we will use the Reddit portal - one of the largest Internet forums of this kind. It also has subforums that allow aggregation of information by topic. This will allow us to retrieve 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 analyse sentiment. It is a model pre-trained on millions of texts and considered a state-of-the-art model in the field of NLP. We then use stock market data, technical analysis indicators and sentiment analysis to try to predict the closing price of a given company the next day. For the prediction of one day we use explanatory variables from the previous 30 days. With the model, we tried to prepare an investment strategy that allows to make buying and selling decisions. We have taken into account transaction costs, the different cut-off points required for a trade and the possibility of potential short selling.

The following hypotheses were verified during our study:

Hypothesis 1 (H1) *Using data from sentiment analysis positively influences the behaviour of the model*

Hypothesis 2 (H2) *The use of data from technical analysis positively influences the behaviour of the model*

Hypothesis 3 (H3) *The investment strategy developed by us based on the model predictions is able to outperform the Buy&Hold strategy*

Hypothesis 4 (H4) *The model architecture can be successfully applied to more than one company*

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

2 Related Work

3 Data and Methodology

3.1 Stock Data

In our analysis we decided to analyze the algorithm on 4 large companies from the gaming industry. This sector was chosen due to the fact that the study takes into account the influence of sentiment analysis - one of our hypotheses is that the opinions of Reddit

users have a particularly significant impact on valuations of a given company in the gaming industry. As many as 4 companies were chosen for the analysis - Electronic Arts, Ubisoft, Take-Two Interactive Software, Activision Blizzard, called in the later part of the paper by their stock ticker: EA, UBSFY, TTWO, ATVI. The selection of more than one company was influenced by making sure that the algorithm was reproducible and universal. Furthermore, attempts were made to use the model learned on one company's data on another company's test data. These companies are among the leaders in the growth industry. The choice of only U.S. companies was due to 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 during replacement, limitations of Reddit's API (Application Programming Interface), and limitations of computing power, the analysis was conducted 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.

Table 1: Descriptive statistics for chosen companies

	Company	ATVI	EA	TTWO	UBSFY
Open	count	7.570000e+02	7.570000e+02	7.570000e+02	7.570000e+02
	mean	6.837792e+01	1.174786e+02	1.434314e+02	1.535412e+01
	std	1.811106e+01	2.062370e+01	3.220382e+01	2.450012e+00
	min	3.956548e+01	7.472812e+01	8.471000e+01	9.050000e+00
	25%	5.210339e+01	9.717636e+01	1.172500e+02	1.393000e+01
	50%	6.857000e+01	1.193464e+02	1.383700e+02	1.548000e+01
	75%	8.241098e+01	1.385884e+02	1.711000e+02	1.681500e+01
	max	1.033197e+02	1.485431e+02	2.104765e+02	2.089000e+01
High	count	7.570000e+02	7.570000e+02	7.570000e+02	7.570000e+02
	mean	6.922366e+01	1.189510e+02	1.453735e+02	1.546871e+01
	std	1.817212e+01	2.062690e+01	3.241062e+01	2.460111e+00
	min	4.119361e+01	7.691432e+01	8.757000e+01	9.270000e+00
	25%	5.284490e+01	9.860734e+01	1.190800e+02	1.396000e+01
	50%	7.004000e+01	1.206978e+02	1.407800e+02	1.564000e+01
	75%	8.327678e+01	1.402643e+02	1.728500e+02	1.694000e+01
	max	1.040263e+02	1.495559e+02	2.149100e+02	2.134000e+01
Low	count	7.570000e+02	7.570000e+02	7.570000e+02	7.570000e+02
	mean	6.738189e+01	1.158599e+02	1.413383e+02	1.521634e+01
	std	1.794660e+01	2.052602e+01	3.182764e+01	2.440169e+00
	min	3.908489e+01	7.344622e+01	8.441000e+01	9.050000e+00
	25%	5.142120e+01	9.571559e+01	1.158500e+02	1.374000e+01
	50%	6.707523e+01	1.170509e+02	1.363400e+02	1.538000e+01
	75%	8.128642e+01	1.370253e+02	1.689600e+02	1.667000e+01
	max	1.020559e+02	1.457801e+02	2.094350e+02	2.067000e+01
Close	count	7.570000e+02	7.570000e+02	7.570000e+02	7.570000e+02
	mean	6.832515e+01	1.174449e+02	1.434253e+02	1.533973e+01
	std	1.802681e+01	2.056176e+01	3.208114e+01	2.456598e+00
	min	3.933990e+01	7.425114e+01	8.463000e+01	9.140000e+00
	25%	5.210339e+01	9.698756e+01	1.176000e+02	1.384010e+01
	50%	6.947362e+01	1.190582e+02	1.381900e+02	1.551000e+01
	75%	8.230000e+01	1.384656e+02	1.706700e+02	1.682000e+01
	max	1.033098e+02	1.482325e+02	2.133400e+02	2.124000e+01
Volume	count	7.570000e+02	7.570000e+02	7.570000e+02	7.570000e+02
	mean	7.852192e+06	3.352595e+06	1.682517e+06	1.603439e+05
	std	4.732384e+06	2.796399e+06	1.258285e+06	3.360945e+05
	min	1.562888e+06	6.060640e+05	2.116420e+05	1.188800e+04
	25%	5.243074e+06	2.068342e+06	1.023114e+06	3.837000e+04
	50%	6.692846e+06	2.723601e+06	1.351017e+06	5.767100e+04
	75%	8.966312e+06	3.757614e+06	1.913890e+06	1.117520e+05
	max	5.170888e+07	3.870450e+07	1.894501e+07	3.997347e+06

3.2 Text Data

Due to the inclusion of sentiment analysis in our study, the source of textual data sourcing had to be chosen. Due to the fact that we have chosen in the study only the companies from the growth industry, we 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 us to obtain data for our study - it is divided into parts, the so-called subreddits, which gather people interested in a particular topic. For our study, we chose the r/Games subreddit. This is the largest forum on this site dealing with the subject of games. There are more than 3.1 million users on it. Next, for each company selected by us, keywords were chosen to retrieve the data. The keywords were the names of the most popular game series of a particular publisher and the publisher's name itself. Next, using the psaw library for python, we found all the comments that had been posted over a predefined period of time on the r/Games subreddit containing the keywords mentioned below

Table 2: Key words chosen for each 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		

3.3 Train test split

To avoid overfitting and evaluate models and investment strategies on data that was not used during training, we decided to split our dataset into train dataset and test dataset. 75 % percent of the dataset has been assigned to train dataset, while the remaining 25 % to test dataset. 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 our findings on data that hasn't been seen by our system before.

3.4 Technical Analysis

3.4.1 Moving Averages

Several types of moving averages are used. The first and simplest of them is a simple moving average (SMA). It takes into account in the same degree all observations in a given time window. For obvious reasons, one may assume that closer dates may have more significant influence on future price. Therefore, in addition to the SMA, Weighted Moving Average (WMA) and Exponential Moving Average 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.

3.4.2 Bollinger Bands

Bollinger Bands consist of three bands. 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.

3.4.3 Bollinger Bands

Moving Average Convergence Divergence (MACD) consists of two lines. 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.

3.5 Data Scaling

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:

$$x_{\text{scaled}} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

3.6 Sentiment Analysis

Sentiment analysis is a process of extracting users' feelings and emotions. It is a part of Natural Language Processing. It boils down to trying to determine with a given probability whether a given statement was positive, negative or neutral. Then such predictions are converted into numerical data. There are many different types of models for sentiment analysis, but the vast majority of them are based on machine learning. In our study, the BERT (Bi-Directional Encoder Representations from Transformers) model created by Google researchers was used.

3.7 BERT (Bi-Directional Encoder Representations from Transformers)

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 a time. 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. Then this model can be trained for any task just by adding one extra layer. Moreover, one can find many pre-trained models ready for download on the internet. This makes it possible for ordinary users without astronomical computational capabilities to use such a powerful model for their tasks by training it only on small amounts of labeled data.

3.8 Sentiment Data Transformation

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 key word related to a given company were combined into one matrix. Their sentiment was then calculated. This data was aggregated by day of creation into the following vector for each day:

$$\text{Sentiment vector for day } i\text{:}$$

$$[n, \mu, \sigma, med, Q_1, Q_3] \tag{2}$$

where:

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

3.9 Basic Recurrent Neural Network

Recurrent neural networks (RNN) is an extended version of artificial neural networks. Its main advantage is having internal memory which allows it to process sequences. Hidden state allows previous outputs to be used as input for further parts of the sequence. This ability to remember previous states makes RNN suitable for time series forecasting.

3.10 Gated Recurrent Unit

Gated Recurrent Unit is an extended version of Recurrent Neural Networks which adress RNN vanishing gradient problem [2]. It's architecture has been proposed by K.Cho in 2014 [3]. Its name comes from gating mechanisms which allow the perceptron to choose which information should be saved or forgotten. It is similar to long 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.

3.11 Convolutional neural network

Convolutional neural network is another class of artificial neural networks used in our proposed GAN model. It is widely used in many different fields, including computer vision, speech processing, and text processing [4]. 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 [5]. It has been chosen for our study due to its good differentiating capabilities.

3.12 Generative adversarial network

Generative Adversarial networks are a family of neural networks first proposed by J. Godfellow in 2014 [6]. 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. They are currently being tested in time series prediction as well. J. Yoon et al proposed timeGAN for generating synthetic time series [7], P. Sonkiya et al proposed S-GAN which is a predictive model [8]. 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 minimise the difference between the true and synthetic distributions, while the discriminator tries to maximise this difference. This can be represented by the following min-max function:

GAN min-max function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3}$$

In classical GAN models, the input to the G generator would be a latent vector derived from N(0, 1). In the model we use, this is replaced by a vector consisting of both the sentiment vector [9], data from the stock market and technical analysis indicators for early convergence [10]. 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. 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 [9][10]. The generator loss function is shown as follows:

Generator Loss Function:

$$-\frac{1}{m}\sum_{i=1}^{m}\log\left(D\left(G\left(x^i\right)\right)\right) \tag{4}$$

The discriminator in our model is a network composed of a 1-dimensional Convolutional

Neural Network. 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. It 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 which flattens the input. It is followed by three multilayer perceptron (MLP) layers containing 220, 220 and 1 units, sequentially [9][10]. The discriminator loss function is shown as follows:

Discriminator Loss Function:

$$-\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(y^i\right)+\log\left(1-D\left(G\left(x^i\right)\right)\right)\right] \tag{5}$$

The optimizer used for both Generator and Discriminator was ADAM and the learning rate was set to 0.0016.

3.13 Evaluation of the model

Evaluation of the model is going to be done using two metrics:

Root Mean Square Error (RMSE):

$$RMSE=\sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i-x_i)^2} \tag{6}$$

where:

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

Mean Absolute Error (MAE):

$$MAE=(\frac{1}{n})\sum_{i=1}^{n}|y_i-x_i| \tag{7}$$

where:

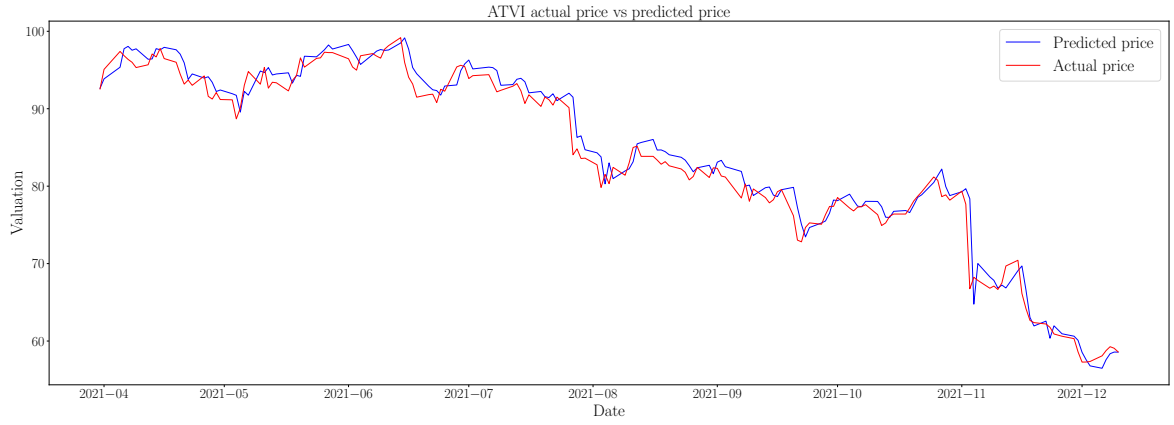
x_i = true value
y_i = predicted value
n = number of observation

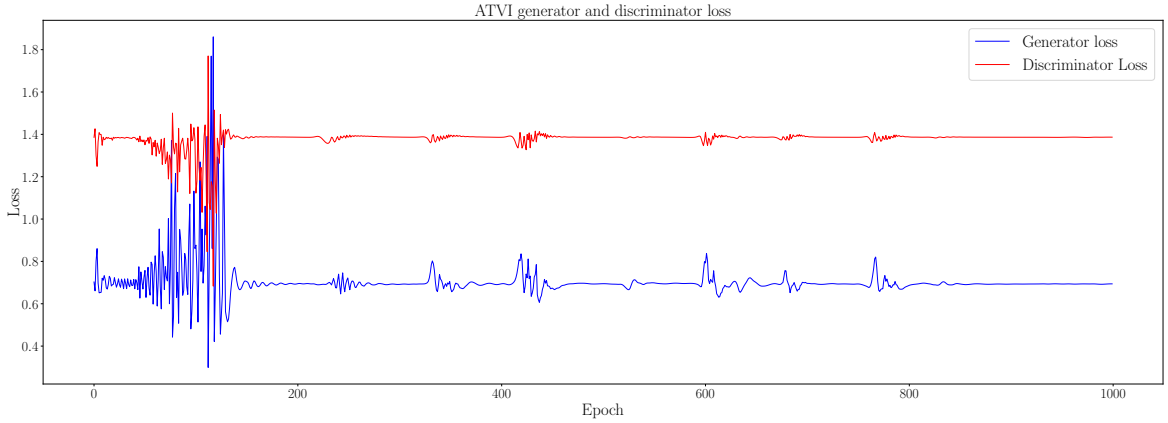
4 Results of the model

The model was trained using the Python language and its frameworks - tensorflow [11] and keras [12][13]. 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 workout was run with an epoch parameter of 1000 and a hyper parameter optimisation was also performed. However, due to the limitation of computing power, it could not be extensive. The behaviour of the network after removing the MLP layers from both the generator and the discriminator was checked. This did not give the desired results, therefore in the further part of the study the parameters proposed by P. Sonkiya et al. were used. The behaviour of the model with different types of input data for the generator was also checked. Both the vector containing sentiment statistics and technical analysis indicators were removed. 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 pollution from reddit. However, this topic needs further exploration with more computational resources. For each of the selected four companies a separate model has been trained for each of the four selected companies.

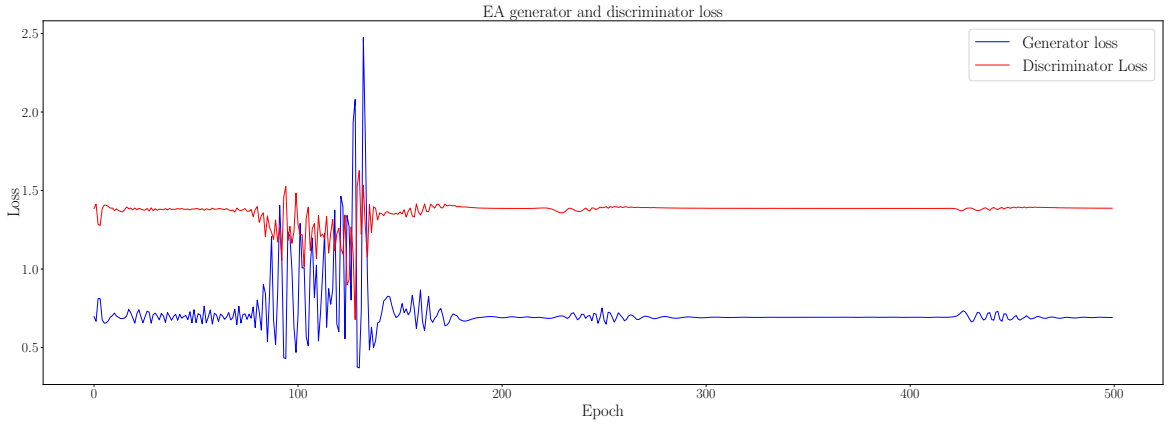
Table 3: Error metrics for chosen companies

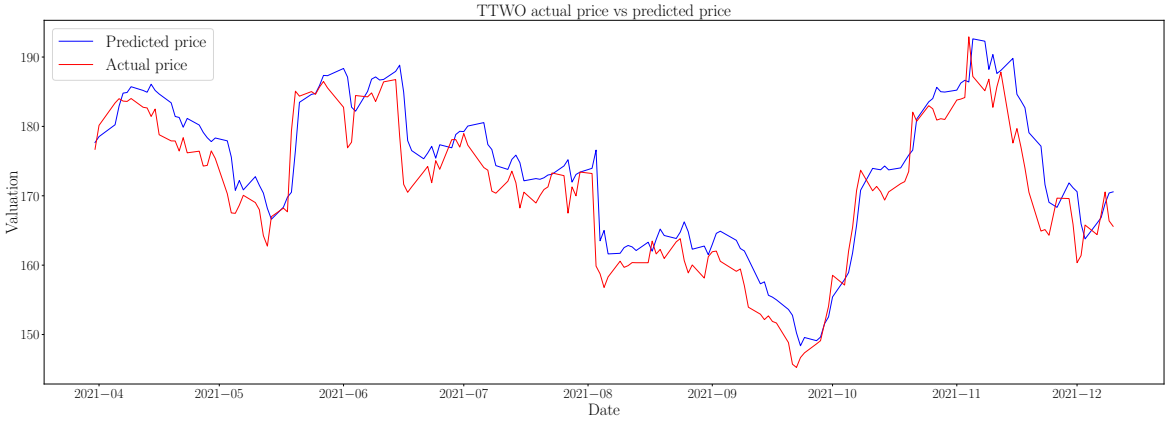
	ATVI	EA	TTWO	UBSFY
MAE	1.358500	1.342700	3.540500	0.257800
RMSE	1.886700	1.895600	4.506900	0.337600
Close Price Mean	83.648342	139.189536	170.877192	12.836576

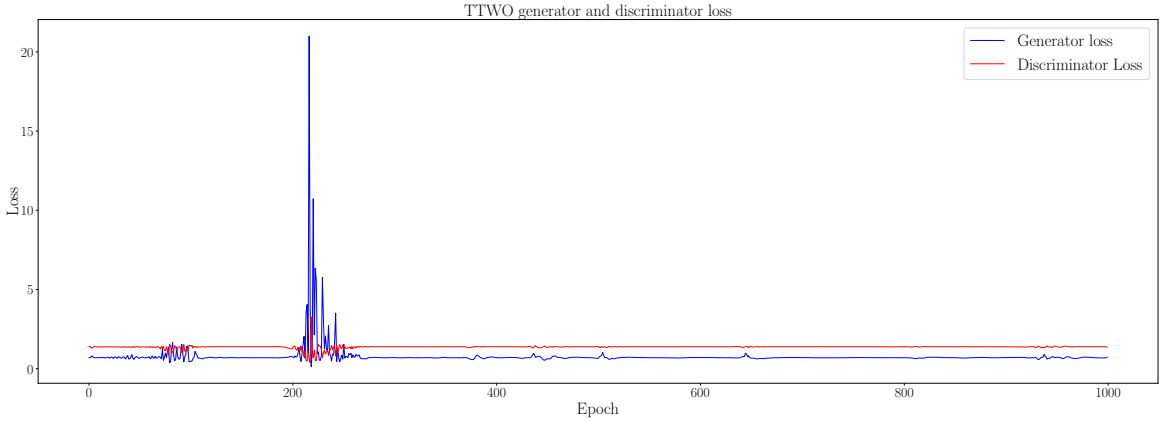


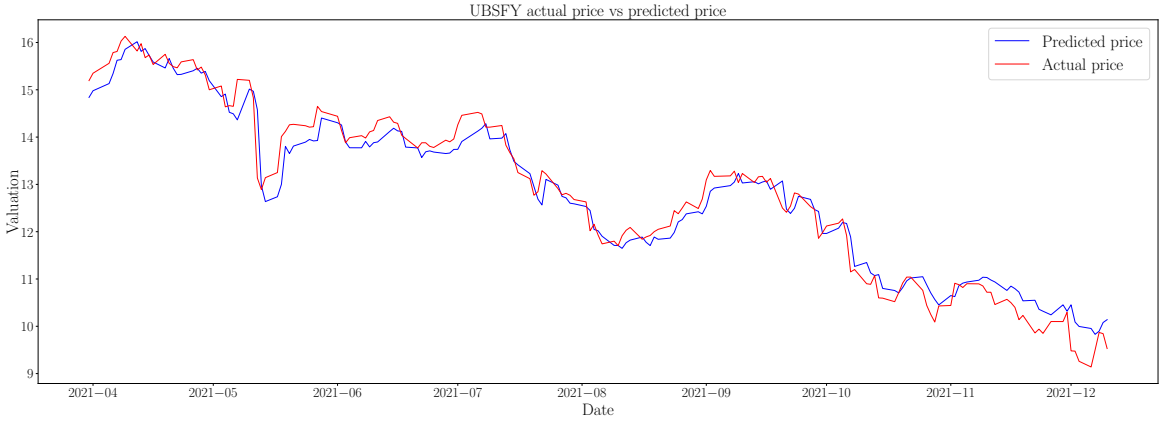

ATVI generator and discriminator loss
Generator loss
Discriminator Loss
Loss
Epoch

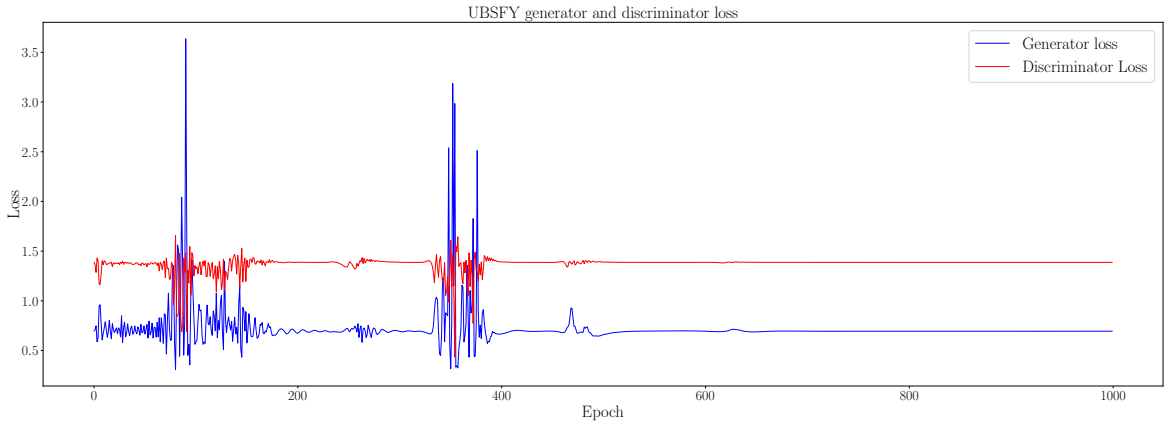

EA actual price vs predicted price
Predicted price
Actual price
Valuation
Date


EA generator and discriminator loss
Generator loss
Discriminator Loss
Loss
Epoch


TTWO actual price vs predicted price
Predicted price
Actual price
Valuation
Date


TTWO generator and discriminator loss
Generator loss
Discriminator Loss
Loss
Epoch


UBSFY actual price vs predicted price
Predicted price
Actual price
Valuation
Date



The results of model analysis on the test dataset are presented in Table 3, where we compare previously selected metrics - MAE and RMSE 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 - MAE equal to 1.5 means something different for the set in which the values of the explained variable are equal to 10 and something different when 150. Due to easy interpretability in the analysis we will focus on MAE metric. For each of the companies, the MAE does not exceed 2.1 % of the average price value over the test period. The best performer here is EA, for which this value is 0.95%. This is interesting as EA was the most thoroughly examined in this work. In the case of this company, the model performed best when the epoch parameter was changed from 1000 to 500. Another important fact is the crossing of the prediction and actual price lines on each of the graphs showing the differences between 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. Such a situation would make it impossible to create a working investment strategy. The loss functions for each of the models are also worth analysing. 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. However, these graphs are different for each of the companies considered in the study and it is difficult to draw generalised conclusions.

5 Investment Strategy

An investment strategy based on the model predictions was also developed in our study. Different versions of it were tested and then the best one was selected for each company. This strategy makes several important assumptions:

- on a given day t we are able to buy at the very end of the day an asset at close price

- *at the start of our experiment we have 1000 dollars*

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

- *we use all our cash resources for each trade*

- *trade fee is equal to 0.0007 $*

The diffrence of prices was calculated during following formula:

$$diff_{t+1} = \hat{y}_{t+1} - y_{t1} \tag{8}$$

where:

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

Then, the buy/selll/hold signal is chosen using following equation:

$$signal = \begin{cases} buy & diff_{t+1} > y_t \cdot \alpha_{buy} \div 100 \\ sell & diff_{t+1} < -(y_t \cdot \alpha_{sell} \div 100) \\ hold & otherwise \end{cases} \tag{9}$$

where:

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

The α parameter can be interpreted here as a cut off point, that requieres the prediction of the price to be bigger then the last price. It is made to assure, that our system makes transaction only when the model is very confident about price change. If that wasn't taken into consideration, transaction costs would have made it unprofitable.

Table 4: Signal hyper parameters

top_cut_off	down_cut_off	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 optimisation 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 maximising 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

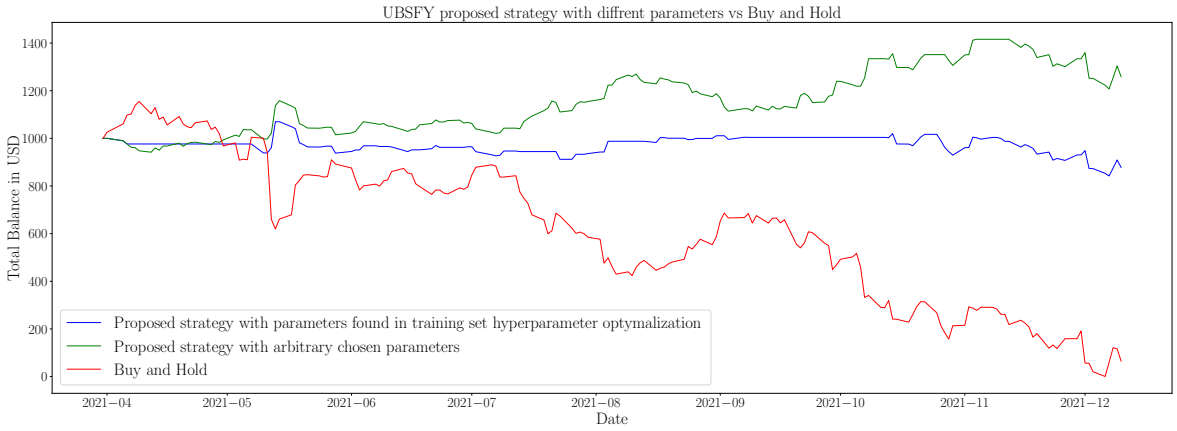
ticker	top_cut_off	down_cut_off	if_short
UBSFY	1.8	1.4	True
EA	2.6	0.6	True
TTWO	1.6	0.4	True
ATVI	0.0	2.2	True

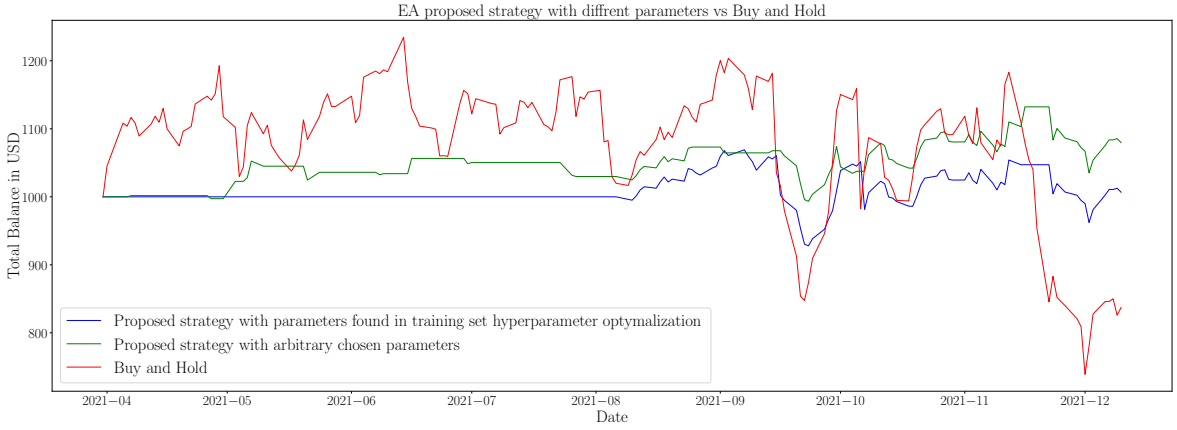
The parameters for the investment strategy found in this way are subject to some bias. It is highly likely that the predictions of the model look quite different for the

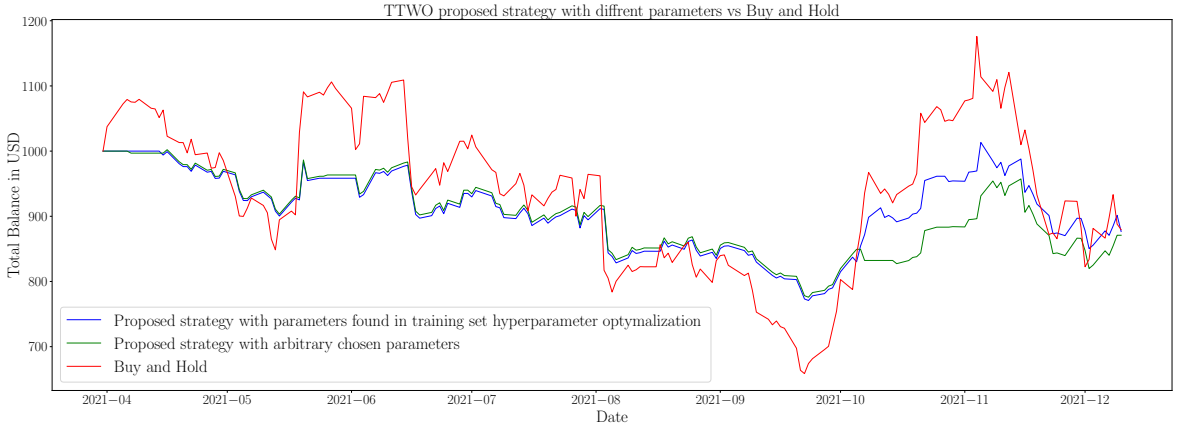
training set on which it was learned than for the test set. Therefore, the fitted parameters may be best for the training set, but not for the data it has not seen. Therefore, an additional set of parameters was arbitrarily chosen. The parameter top-cut-off is equal to 2, down-cut-off to 0, and if-short to True. A high top-cut-off is aimed at a conservative approach to buying, making trades less frequent and therefore our funds are not eaten up by transaction fees. A zero down-cut-off will instead give an instant sell if the model prediction is higher than the previous price. This is intended to minimise losses. A short selling option has also been made available.

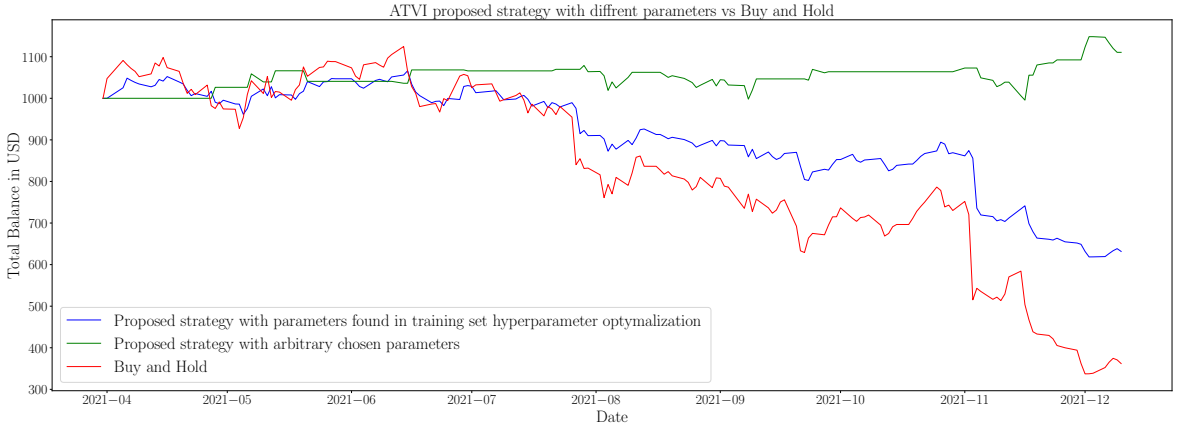
Table 6: Diffrent strategies balance at the end of test time peroid

ticker	Buy and Hold	Proposed strategy with parameters found in hyperparameter optymalization	Proposed strategy with arbitrary 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 $




EA proposed strategy with diffrent parameters vs Buy and Hold
Total Balance in USD
Date
Proposed strategy with parameters found in training set hyperparameter optymalization
Proposed strategy with arbitrary chosen parameters
Buy and Hold


TTWO proposed strategy with diffrent parameters vs Buy and Hold
Total Balance in USD
Date
Proposed strategy with parameters found in training set hyperparameter optymalization
Proposed strategy with arbitrary chosen parameters
Buy and Hold


ATVI proposed strategy with diffrent parameters vs Buy and Hold
Total Balance in USD
Date
Proposed strategy with parameters found in training set hyperparameter optymalization
Proposed strategy with arbitrary chosen parameters
Buy and Hold

As can be seen in the charts above, in 3 out of 4 cases the strategy we presented outperformed the Buy and Hold strategy. Interestingly, the arbitrarily indicated parameters for the investment strategy turned out to be better than those chosen as a result of 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. 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 the worst and needs further research. It is important to note that each of these companies was in a downtrend. The Buy and Hold strategy has in each case recorded losses. This makes it all the more optimistic that 3 of the 4 assets made significant gains.

6 Conclusion

The aim of our work was to test the feasibility of using the GAN network and the BERT model to predict the valuation of four listed companies. The BERT model was used to analyse the sentiment of comments about a given company and its games from the reddit.com portal. 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 our model, we created a system to decide on a buy or sell signal. An optimisation of the hyperparameters in this system was carried out in order to find profit-optimising parameters. For 3 of the 4 companies we managed to create a strategy that significantly outperforms the buy and hold approach.

The study tested the hypotheses raised in the introduction.
The first hypothesis *(H1) Using data from sentiment analysis positively influences the behaviour of the model* has to be rejected since we haven't managed to create well performing model without throwing sentiment data away. Therefore we have to reject H1
The second hypothesis *(H2) The use of data from technical analysis positively influences the behaviour of the model*, holds due to the fact that technical analysis improved every of our models.
The third hypothesis *(H3) The investment strategy developed by us based on the model predictions is able to outperform the Buy&Hold strategy* holds for 3 / 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 proven that our system can outperform buy and hold approach, H3 cannot be rejected.
The fourth hypothesis*H (H4) 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 / 4 tested companies.
Finally, the fifth hypothesis *(H5) Adding the possibility of short selling to the investment strategy increases the profitability of our system*, holds due to the fact that that strate-

gies with ability to short selling performed better

7 Intentional self-criticism

One 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 characterise the best model. This is also a considerable difficulty in optimising 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 behaviour and different results with the same parameters. They become deterministic only after overtraining when all parameters are fixed. This effect is further compounded by the fact of using two networks competing with each other and having separate two loss functions. Further analysis can focus on stabilising 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 minimise the risk of loss in the event of a sudden unforeseen 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] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 2014

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

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

[4] L. Alzubaidi et al, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, 2021, 14

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

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

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

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

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

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

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

[12] F Chollet et al, Keras, 2015

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

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