DEEP LEARNING APPLIED TO PUBLIC COMPANY VALUATION FOR VALUE INVESTING

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Abram Paul Haich

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Computer Science and Statistics

April 2021

Fargo, North Dakota

North Dakota State University
Graduate School

Title

DEEP LEARNING APPLIED TO PUBLIC COMPANY VALUATION
FOR VALUE INVESTING

By

Abram Paul Haich

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

NAME OF DEGREE

Master of Science in Computer Science and Statistics

SUPERVISORY COMMITTEE:

Dr. Juan Li
Chair

Dr. Gang Shen

Dr. Kenneth Magel

Approved:

04-30-2021
Date

Dr. Saeed Salem
Department Chair

ABSTRACT

Value investing is an investing approach that seeks to discover and take advantage of price discrepancies between the market price and the actual value of a company (intrinsic value). The purpose of this work is to measure the intrinsic value of companies using an approach that has had success in the broad field of Artificial Intelligence, Deep Learning. Finding patterns in large amounts of data is what Deep Learning can be used for. Typically for value investing an investor will seek to find conservative estimates on the current value of a company by analyzing fundamental data. Our method attempts to perform these estimates in a data driven manor using Deep Learning to estimate the intrinsic value of a company with the overall goal of aiding the Investor in uncovering undervalued companies.

DEDICATION

To my wife who has helped me endlessly during my master's program and is there for me in every step of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

ANN..Artificial Neural Network

AI ....Artificial Intelligence

CNN ....Convolutional Neural Network

DCF...Discount Cash Flow

EPS..Earnings Per Share

FFN ....Feed Forward Neural Network

MSE ....Mean Squared Error

PE Ratio ....Price to earnings ratio

RNN ....Recurrent Neural Network

S&P ....Standard and Poor Index

LIST OF APPENDIX TABLES

1. INTRODUCTION

1.1. Overview of Contributions and Approach

The purpose of this work is to predict the future cash flows of a public company using Deep Learning, with the overall goal of arriving at a price one should pay for the company at present. This price could be different than the current market price, this is the premise of value investing as we will see. In our research section we explain that the common approach to the valuation of public companies is performed with simple linear growth models looking at historic earnings and price to earnings ratios to predict future trajectories. The work and novelty of this project is to leverage Deep Learning to replace these simple linear models with a more non-linear approach that can provide better results. It is difficult to find papers that performed similar approaches due to the high degree of expertise in software engineering required to combine, aggregate, preprocess and normalize the data into something meaningful for the model. For example, to use multiple datasets as was done in this work requires a custom library to be built so that one can bring in valuable sector and industry information. The Compustat Database also requires working knowledge of SQL to properly interact with it. There are funds that pay millions of dollars per year for datasets with this information so that it can be easily digested (Fridman, 2021). Another reason it's hard to find similar work in this area is that company valuations can be difficult to measure. To make sure our model is performing well on the dataset a novel approach is proposed. The real future cash flows in our dataset of a company will be discounted to produce an actual valuation. It is the goal of our deep learning model to predict this valuation. We will then compare our model to a simple linear growth model routinely used for valuing companies. This will help arrive at an overall performance benchmark for the model.

1.2. Value Investing

Benjamin Graham is considered the “father of value investing” (Graham, The Intelligent Investor Rev Ed, 2003). In his seminal work “The Intelligent Investor” Graham defines the approach of value investing. His main observation is that in the short term the market behaves like a voting machine, changing price to reflect daily news articles and overall investor moods. However, in the long term it behaves more like a weighing machine; slowly aligning price and underlying value of a company.

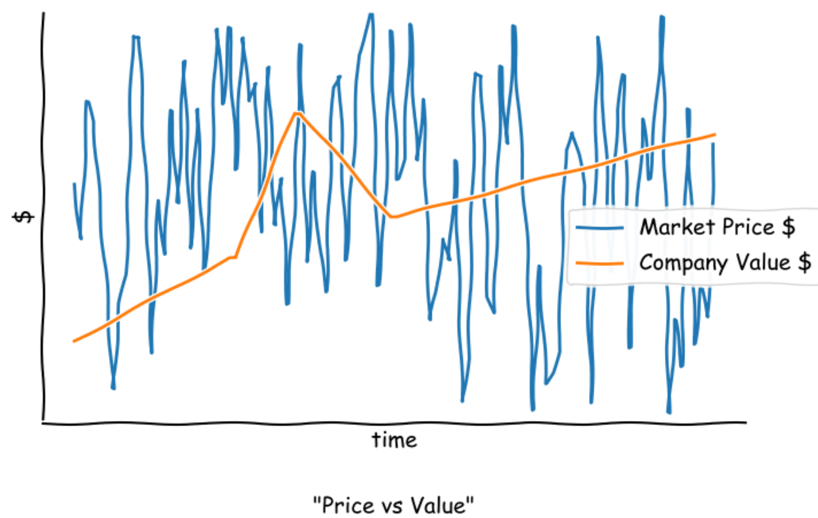


Figure 1. Demonstration of price and value in short-term vs. long-term.

The value investing approach uses fundamental analysis to study income statements, balance sheets and cash flow statements with the goal of determining a current price for a company. Graham’s guiding principal regarding value investing is that investors should seek investing as though they are buying a whole company and then determine how much they should pay for it. This means that the prospective investor should only consider what he/she would pay for the company as a whole and then divide that value by the number of shares.

In Benjamin Graham's second book "Security Analysis", Graham outlines a formula for predicting intrinsic value of a stock. Intrinsic value being not what the market would pay for a company but what it is intrinsically worth given its cash flow and assets. If this number is greater than the current market price by an amount called the margin of safety, then the stock should be purchased as the market is selling it at a discount. Graham's formula for intrinsic value is outlined in Equation 1.

$$\text{Intrinsic Value} = \frac{\text{Earnings}}{\text{Shares}} * \left(8.5\text{g} + 2(\text{Company Growth Rate})\right) * \frac{4.4}{\text{AAA bond rate}} \qquad (1)$$

This formula attempts to evaluate a company against risk free investments, future company growth rate, and previous moving average of earnings/share for a company. In Graham's time this formula worked well as some companies were selling below their book value. But as these "bargains" disappeared, and as the market saw more growth companies in the exchanges (companies selling at very high price to earnings ratio) Graham's formula became obsolete and less useful. However, the main premise of value investing was still valid and profitable, with slight changes to Graham's approach Warren Buffet (one of Graham's students) became the most famous and rich value investor.

Warren Buffet, considered the greatest investor of all time, is one of Graham's students who used the value investing approach to build a multi-billion-dollar company Berkshire Hathaway. His approach differs slightly than Graham's and is an approach we will take in our research. Graham was focused on companies called "cigar butts" (Schroeder, 2008). These companies are called "cigar butts" because they are companies with one last puff of smoke in them. Typically, this last puff is a liquidation sale of assets or some company split. When Buffet started out, he performed well by investing in these types of companies. However, when he went into a partnership with Charlie Munger his investment approach changed. Instead of finding these “cigar butt” companies at rock bottom prices he switched to finding companies that contain moats selling for reasonable prices. By moat, Munger means that there is a barrier to entry for other companies to enter in the same area. For example, Coca-Cola has a secret recipe that is hard to replicate by competitors. The purpose of the Deep Learning model we develop is to estimate the reasonable price aspect of this formula. Estimating a moat is a subjective thing requiring a high degree of knowledge in an area (investors have specific areas of expertise that they focus). However, monitoring business health from numbers and arriving at fair prices is possible for a Deep Learning model and so that will be the focus of this work.

1.3. Discount Cash Flow

Discounted Cash flow (DCF) analysis is a valuation technique commonly used in valuing companies. It’s a method that looks at cash flows over a period discounting them to the present value. It runs on the premise that a dollar today is worth more than a dollar tomorrow which is referred to as the time value of money. Consider denoting cash flows of a company as FCF_1, FCF_2, ... Then by applying a DCF to these cash flows we arrive at:

$$V_0 = \frac{FCF_1}{1+k_0} + \frac{FCF_2}{(1+k_0)(1+k_1)} + \cdots + \qquad (2)$$

Where k_0,k_1,k_2 ... are the cost of capital during the periods $0,1,2$... These costs of capital could be considered the risk-free rate of returns like a bond or a savings account (Lutz Kruschwitz, 2020). For our work we contrast our risk-free rate as the returns on S&P index over 5-year periods which is about 10%. The reason we use the discount cash flow method in our deep learning model is due to its robust formulation of actual future cash flows and time value of money considerations it provides. Additionally, it is suggested that Warren Buffet uses a similar approach when evaluating companies as mentioned in his letters from (Buffett & Fund, 2014).

1.4. Deep Learning

To understand Deep Learning, it is important to first understand Artificial Intelligence (AI) and Machine Learning. AI is a field of Computer Science focused on emulating human intelligence in machines. Machine Learning is a subfield of AI focused on making machines that can learn from data. Deep Learning is a subfield of Machine Learning that uses Artificial Neural Networks (ANNs) of more than two layers to learn from data (Bishop, 2006).

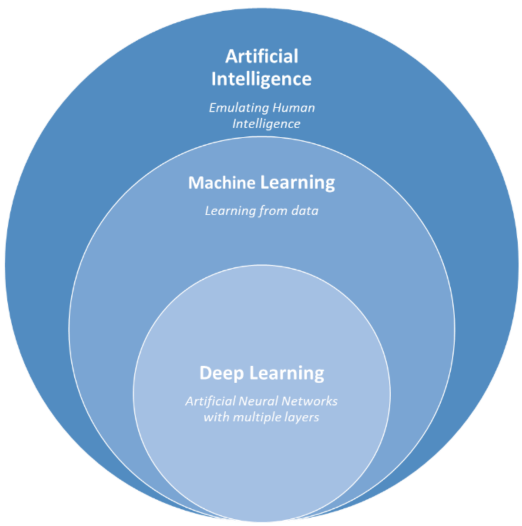
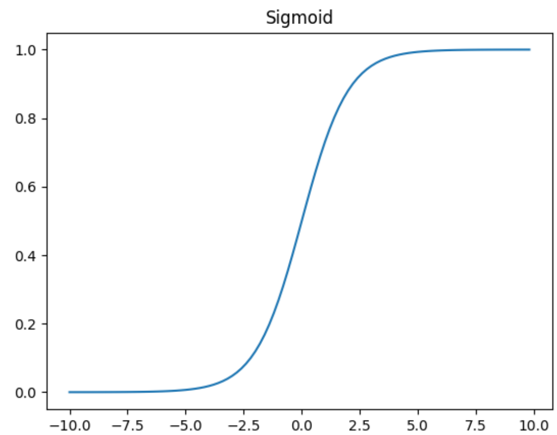


Figure 2. Relationship of AI, Machine Learning and Deep Learning.

ANNs themselves are made up of a simple building blocks called neurons. A single neuron has inputs that get multiplied by weights and summed together along with a bias. This value is then fed into an activation function. Activation functions are chosen based on aligning use cases to specific properties of an activation function. Mathematically for neuron z_i we have:

$$z_i = \emptyset\left(w_{0,i} + \sum_{j=1}^{m} w_{j,i} * x_{j,i}\right); \text{where } \emptyset \text{ is some activation function} \qquad (3)$$

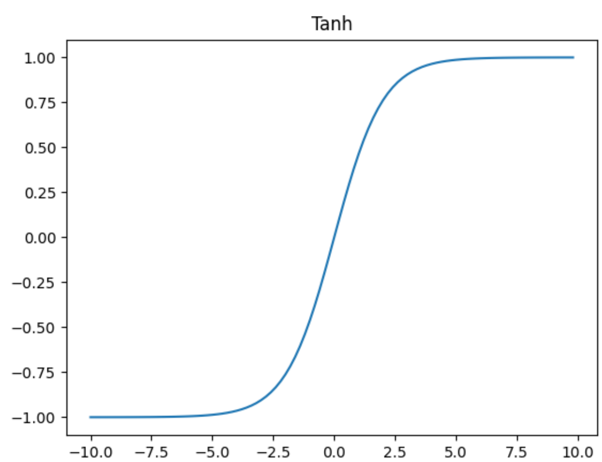
Activation functions are where the true power of neural networks come into play. Without activation functions ANNs would be linear models. Adding activation functions provide non-linearity to the model allowing them to better capture higher order functions. One of the most common activation functions in Neural Networks is a sigmoid. Sigmoid activation functions force the output values between the range 0 and 1. This can be useful for describing probabilities or for classification models with only two classes.



$$f(x) = \frac{1}{1 + e^{-x}}$$

Figure 3. Graph of sigmoid activation function.

Another commonly used activation function is the hyperbolic tangent (tanh) function. This function forces the output between -1 and 1. This can be useful for modeling normal distribution where the -1, and 1 could be considered differences from mean of 0.



$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Figure 4. Graph of tanh activation function.

Neurons can be chained together to build a simple network with different layers. See Figure 5. This particular network architecture is called a Feed Forward Neural network (FFN) and although it was proven to be able to approximate any function, this is terribly difficult in practice due to gradient optimization (Goodfellow, 2016). Over time the community migrated away from only FFNs to more specialized network architectures. For example, in image processing a more specialized network architecture is called a convolutional neural network (CNN). Whereas in the time series domain a contrasting specialized network architecture used is the recurrent neural network (RNN). The reason these perform better than a FFN in these domains is because they both allow better modeling of their corresponding domains with less weights and easier to optimize gradients.

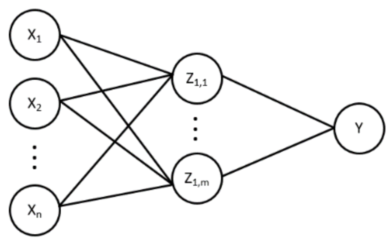


Figure 5. Diagram of a simple Multilayer Neural Network

As mentioned in previously, when modeling time series models, like stock market data, it is a common practice to use Recurrent Neural Networks. These are neural networks that have feedback loops to allow previous outputs to influence current time step inputs. One type of RNN used commonly are Long Short-Term Memory (LSTM) models. LSTM models contain "memory cells" that can carry important information from a particular point in time forward for the next time sample. In order to properly maintain a valid state within these cells a forgot gate is used to remove irrelevant information and avoid vanishing gradients (exceptionally small gradient that can make updating network impossible) that can occur in typical RNNs (Sepp Hochreiter, 1997). Let x_t be independent data we are feeding into the LSTM. Consider Figure 6, as an input x_t is fed into the network, a forget gate is calculated using a sigmoid activation function which pushes the value of the output between 0 and 1. This could be considered a FFN since we have weights, bias and activation function. The subsequent step is to update the cell state which contains the information we would like to remember. This process contains another FFN with output between 0 and 1 and additionally the state itself which is a FFN with a tanh activation function. The next step in this process is to update the previous cell state (C_{t-1}) with the new information provided by the previous steps. This new value C_t is then output from the LSTM block. The last and final step in the LSTM is to output the predicted value using the current cell state with the current input value. Although a bit more complicated than a regular RNN these models solve the vanishing gradient problem by systematically having a mechanism to remember the important parts of a sequence and forgetting the least important parts.

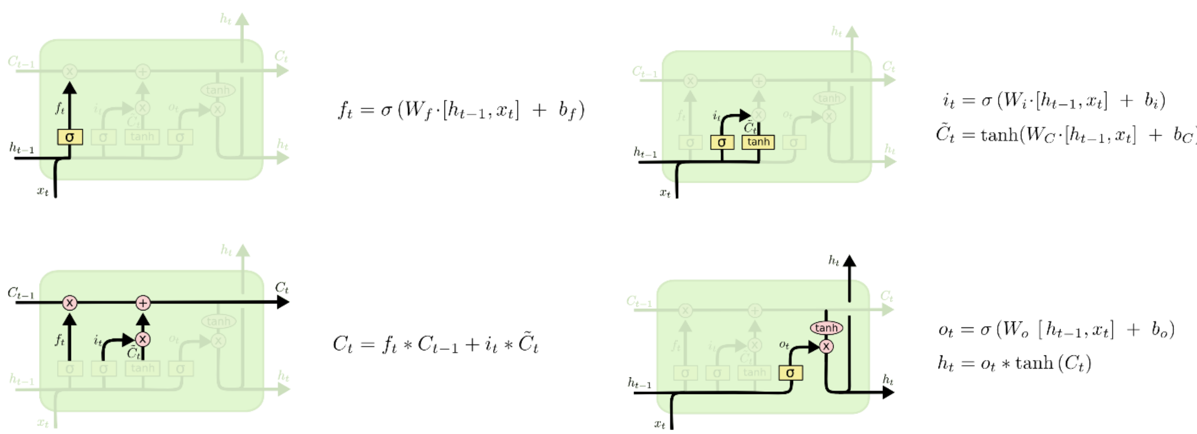


Figure 6. LSTM Network. (oinkina, 2015)

2. RELATED WORK

2.1. Value Investing Approaches

In his book “Rule #1: The Simple Strategy for Successful Investing in Only 15 Minutes a Week!” Phil Town outlines his approach to the valuation of a public company. On his website you can find several calculators to calculate the Compound Annual Growth Rate (CAGR) for EPS, market capitalization, PE Ratio Growth Rate and finally a calculator to discount these predictions back to present value (Town, ruleoneinvesting, 2021). His company valuation is called the “Sticker Price” and forms the basis for if you should buy the investment. Buffet performs similar analysis as mentioned in (Buffett & Fund, 2014) so we will use this method as our baseline. There are problems with this approach that we would like to correct though. The first being that his method is linear, i.e. his method just takes historical growth and multiplies by the number of years out one would like to predict. However, companies typically fail or grow exponentially so this doesn’t capture the true value and curves of these variables. Secondly his approach only considers PE ratio and Earnings to predict PE ratio and Earnings, however many different factors can drive these two variables. By including more variables into a model, the goal would be to get higher accuracy so that one can make more informed decisions when investing.

John Alberg and Michael Seckler are two prominent investors that apply Machine Learning to the Value Investing approach. On their website of their company Euclidean Technologies, they outline their framework for using Deep Learning to value companies using a subset of the data we use in our work (Alberg & Seckler, 2021). The approach they take is largely different then our own but worth mentioning. Their work uses Deep Learning to predict a discrete binary classification. It will predict a 1 if the company under consideration has a higher return on investment than the median return of all companies in one years' time. If it is below this median 0 is output from the model. While their results are very promising, there are a few arguments and criticisms regarding this approach from a value investing standpoint. Firstly, one year out is not that far out in investing terms, value investors typically look 5 years out or more. Secondly their ground truth doesn't consider company performance. Since price doesn't always reflect value, they are instead predicting only price movements not directly reliant on company performance. For example, a popular company could have a higher price relative to the median price but not perform very well and could fall later due to the market realizing the discrepancy between value and price. A great example of price and value discrepancy can be found in the tulip craze of the 1630s. During this time tulip prices went so high they were more valuable than houses. Rare bulbs with certain patterns sold for astronomical prices and they couldn't keep up with demand. Within the span of a year though the bubble had burst causing people to throw away their bulbs for nothing since they were practically worthless due to the oversupply of people selling to get any value they could out of them (Pollan, 2002).

Other research methods similar to our approach for value investing are papers around predicting EPS (Elend, Tideman, Lopatta, & Kramer, 2020). However, these approaches only look a quarter out which is not long term enough for the typical value investor. We do train a model like this though as a baseline for our own method.

2.2. Discounted Cashflow Approaches

An approach that applies Deep Learning to a discount cash flow model similar to how we are performing it is "Deep-Learning the Cash Flow Model" (Clayton, 2020). This paper uses Deep Learning to predict future cash flows of a company. Their model even looks 5 years out for companies to value them at present values as our approach does. The variables they used in their valuations were different than ours and not compared to baseline methods though. Further differences were in the goal of their work. The goal of their work was to use the learned discounted cash flow to support financial planning, cash flow testing, and asset liability management (ALM). In short, they are looking at their research from a finance perspective of an individual company not an investing and deep learning perspective. In our work we seek to extend their work to EPS and PE Ratio predictions so we can compute the value of a company from the investing and deep learning perspective. We then wish to compare our results to not only actual cash flows but moreover the Rule 1 Investing method. This will help us form a robust baseline comparison of performance. One last major difference between their work and ours is in the difference of data. Since a large dataset was created with ours by merging multiple sources, we have a unique approach to the analysis we can perform on company and sector level analysis.

3. METHODS

3.1. Dataset

The data used in this paper was collected from two sources Yahoo Finance and Compustat. The dates range from 1962-2020 and include most publicly traded US stocks/companies (6,586 companies). Price data is of daily occurrence, fundamental data is quarterly. Data was combined and organized into a large dataset spanning many years and many companies. A randomly selected subset of 20% of the companies where chosen for model evaluation/validation and were not trained on. Scaling and normalization of the data was performed by first dividing each company by its enterprise value and then using standard normalization methods:

$$y = \frac{x-\mu}{\sigma} \tag{4}$$

The data utilized was fundamental data such as Income Statement, Balance Sheet and Cash Flow Statement. Caution was taken to get the exact date when these statements were publicly made available in order to avoid making an unrealistic model as outlined in (Prado, 2018). The data field to accomplish this was not at all obvious in the Compustat database. Additionally, things such as annual financial data required merging with quarterly data to arrive at correct public release date. Another important normalization technique that had to occur before being able to use the Compustat data is to adjust for stock splits and other events that can cause any dramatic changes in the number of outstanding shares. This can be achieved by multiplying by an adjustment factor (adjex) from the Compustat Database. Missing data was either forward filled if possible (carry a value forward in time) or entered as a 0. Early companies had more issues with missing data fields then later companies due to financial regulation early on in US History.

Several features were used as inputs to our model as well as for deriving other features such as enterprise value. Those features were Specifically speaking current inputs are:

- Revenue
- Cost of goods sold or COGS
- General and administrative expense or SG&A
- Retained earnings (EBIT is yearly)
- Earnings before interest and taxes or EBIT
- Net income
- Common shares outstanding
- Cash and cash equivalents
- Accounts receivable
- Inventory
- Property plant and equipment or PP&E
- Other current assets or OCA
- Debt in current liabilities
- Accounts payable
- Other current liabilities
- Total liabilities

NOTE: For a more detailed breakdown of variables from Compustat see appendix.

For the first model, the timeseries fundamental data were split into 8 quarters (2 years) to predict next quarters earnings. This earnings estimate was then compared with the actual earnings to arrive at a loss with the overall goal of predicting a company's earnings one quarter out. This helps with evaluating a company since earnings are directly related to the valuation of the company. For example, using these earnings estimate a valuation can be estimated by using a conservative PE ratio for the company's sector. Another way to use this earnings estimate could be by directly using Graham's formula. In Graham's Formula this estimate could provide a better estimate of growth for earnings per share instead of trailing 12 months average.

$$\text{Valuation} = \frac{\text{predicted earnings}}{\text{shares}} * \frac{\text{price sector}}{\text{earnings sector}} \tag{5}$$

For the second model the time series fundamental data were split into 20 quarters (5 years). A derived column was added based on sector PE ratio multiplied by the company's earnings for the quarter. This was to highlight if a company should be paid a premium given that its earnings could be above the mean sector PE Ratio. The difference between this value and next quarters value was then added to make this a cash flow type investment. After this change these cash flows were then summed using the discount cash flow method to arrive at an intrinsic value for the company.

$$\text{Intrinsic Value} = \sum_{k=1}^{20} \frac{\text{return}_k}{(1+0.10)^k} \tag{6}$$

3.2. Models

For both models the same network architecture was used (See Figure 7). The only difference between the two models were modifications on the input data as well as output data for the different features, goals and timesteps. The models were written in PyTorch and many different hyperparameters were chosen before arriving at this final one. i.e. multiple hidden layers, larger/smaller hidden sizes. These settings provided optimal learning rates for the dataset used.

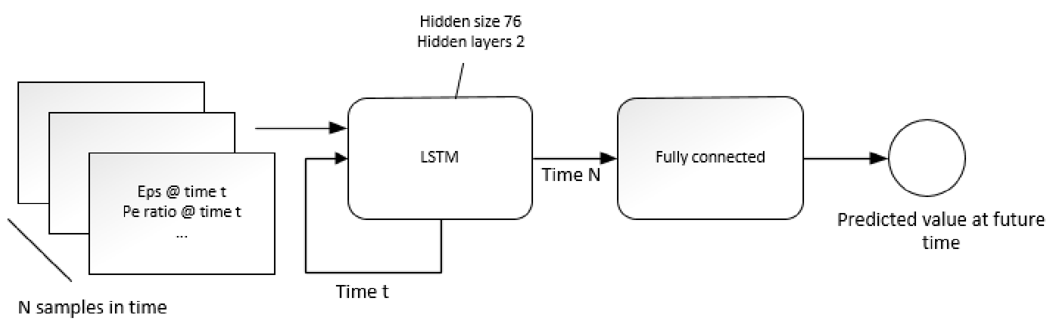


Figure 7. Network architecture.

The first model used was focused on predicating earnings per share. *Earnings Prediction with Deep Learning* (Elend, Tideman, Lopatta, & Kramer, 2020) a recent research paper was able to perform very well with comparable data, so the thought was this would be a good starting point for research. The basic idea of this approach is that predicting future earnings could allow for a better measurement of a company's valuation either using DCF or Graham's formula. An additional goal of this model was to test the validity of valuing companies using the data sources collected before moving on to larger time horizons and more complicated data curation. The data used in this case was quarter fundamental data with no sector/industry information. The model would take 2 years of data in and predict 1 quarter ahead for earnings per share (EPS). It took several different runs to arrive at correct scaling and hyperparameters for correct loss decrease. Several hyperparameters and data curation methods were tried, such as min/max scaling, no scaling, no normalization, normalization, Stochastic Gradient Decent, Adam optimizer. The final hyperparameters and data scaling were enterprise value scaling (divide by enterprise value of company), standard normalization, and Adam for optimization with a learning rate of 0.0001.

The second model was built to predict all future cash flows 5 years out as seen from earnings and sector PE ratio. This model was predicting the following:

$$\text{Intrinsic Value} = \sum_{k=1}^{20} \frac{\text{eps}_k * \text{pe\_ratio\_sector}_k - \text{eps}_{k-1} * \text{pe\_ratio\_sector}_{k-1}}{(1+0.10)^k} \tag{7}$$

The reason a PE ratio for the sector was used was because individual company prices do not reflect the evaluation of the company by premise of value investing. We used the PE Ratio of the sector to compare one company relative to its peers. This avoids issues where a particular company might be favored in the market relative to its peers for no reason other than speculation. The data used in this model was slightly different then EPS model in that sector information was important, so every feature was doubled to include sector average in addition to company under consideration. The time horizon was also larger, 5 years out vs. 1 quarter out so it required more history to be fed into the model (5 years or 20 quarters). Scaling was additionally different for this model. Scaling was performed by performing min/max scaling by year. This was done to help deflate future inflated prices in future times.

3.3. Comparison to Existing Approaches

With our models described and our dataset outlined it is worth mentioning the unique approach we are seeking in this work and how it compares to other work. For our work we are using Deep Learning to predict Equation 7 with the goal of valuing a company through future cash flows. Let's compare this approach to a very similar and common approach for Value Investing. The closet work to ours is the Rule 1 approach. Let us now work through how one can setup the Rule 1 approach to compare to our model. We are looking at predicting 5 years out for our model so we will do the same with this method. Also, we will consider 5 years of historical information. So, using this method we first collect 5 years of historical information for the EPS for the company, and PE ratio for the sector. Using the Compound Annual Growth Rate (CAGR) formula one can arrive at the annual rate of growth for these values see Equation 8.

$$\mathrm{CAGR(EPS)} = \left(\frac{\mathrm{EPS}_{\text{end value}}}{\mathrm{EPS}_{\text{begin value}}}\right)^{\frac{1}{5}} - 1 \tag{8}$$

Using these two values one can predict the future cash flows (EPS rate * PE rate) of a company by taking these two growth ratios and calculating out 5 years into the future as is done in Equation 9. Note the divisor; This is performing the discount cash flow method with minimum rate of return of 10% as we perform in our model. However, there are sever limitations to this method. The first being this approach only considers PE and EPS to predict future PE and EPS.

$$\text{Sticker Price} = \frac{\mathrm{PE}_{\text{current}} * \left(1+\mathrm{CAGR(PE)}\right)^{5} * \mathrm{EPS}_{\text{current}} * \left(1+\mathrm{CAGR(EPS)}\right)^{5}}{(1+0.10)^{5}} \tag{9}$$

However, there are many factors that can affect these so a more appropriate model should consider all fundamental items from the income statement, balance sheet and cash flow statement. The second problem with this approach is that it is a linear method. i.e. it looks at finding slope (annual change CAGR) and uses this slope to predict 5 years out. However, when looking at growth models or hyped companies the growth and decay is typically exponential not linear. In using Deep Learning our hope is that our model will be able to learn these non-linearities and better model the current value of a company.

4. RESULTS

4.1. EPS Model

The earnings per share model reached minimum error around 60 epochs with a MSE under 0.0002.

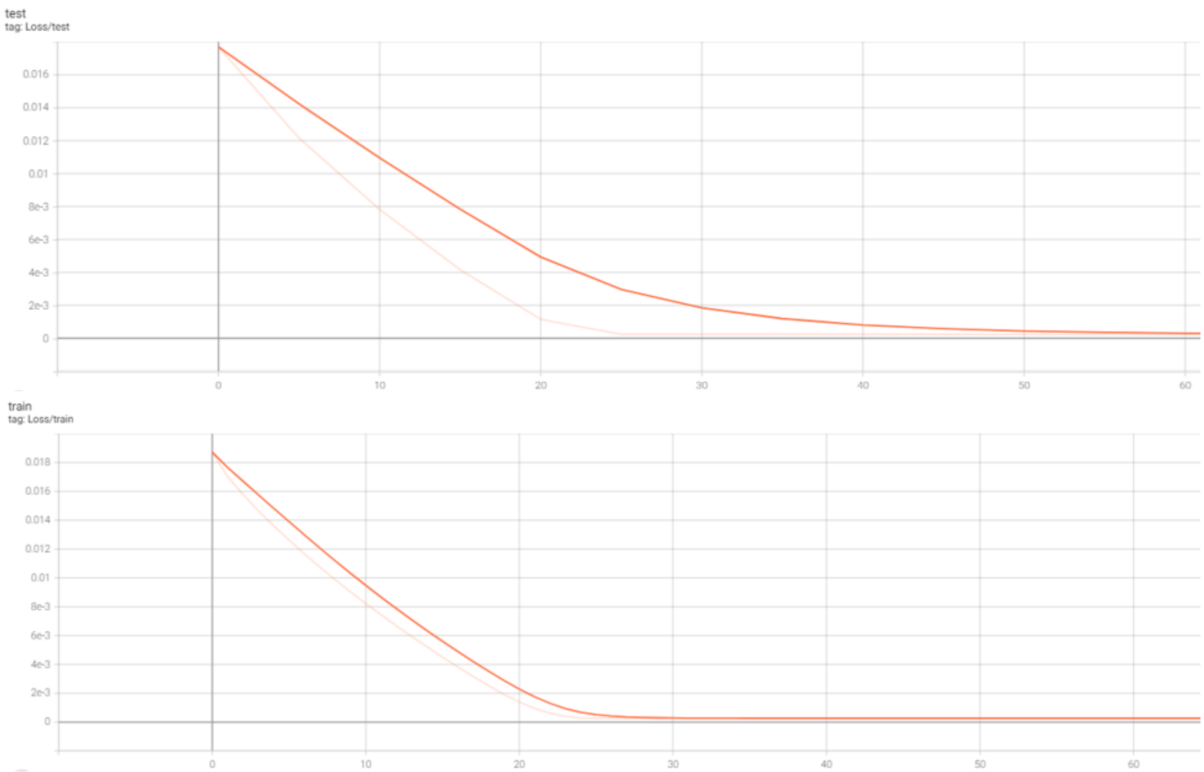


Figure 8. Loss curves for EPS model.

Although the residuals of error had outliers that were drastically off relative to the actual EPS; these residuals were centered around zero and were relatively normally distributed. One outstanding result of this model is its ability to predict spikes in earnings per share. This could be used for instances of options trading where you are predicting a company will lose/gain value on the next quarters earnings release. After this model showed promise the second model was pursued to better evaluate a company's valuation using future cash flows.

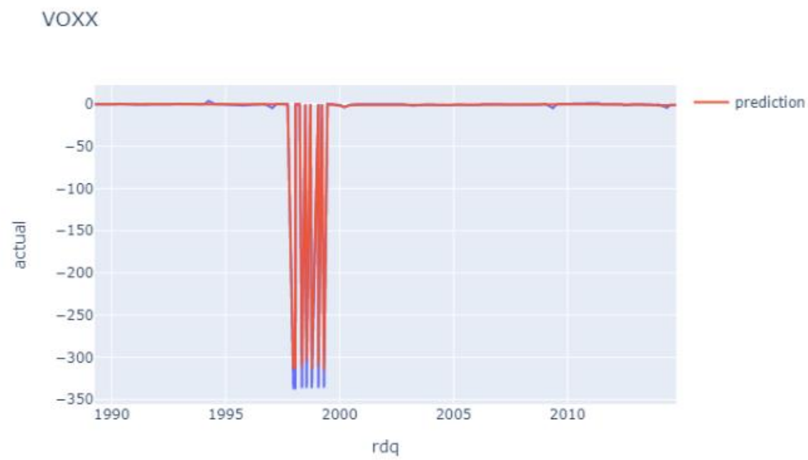


Figure 9. EPS from test set (for VOXX ticker).
Note: Blue is actual, orange is predicted. The sliding window of 2 years was continually shifted and compared to actual earnings per share to generate this plot.

4.2. Valuation Model

The direct DCF valuation model stopped improving around 90 epochs with a MSE under 0.7. Notice that this is a dramatically higher number then the EPS model, this is since the loss is on valuation of a company which can be larger than the EPS since company evaluations can be larger than EPS. Analyzing the percent error on the test set of prediction vs actual valuation leads to 80% of the points falling within the -100 to 100 percent range. With 90% if those in the range of 40% to -40% (see Figure 10)

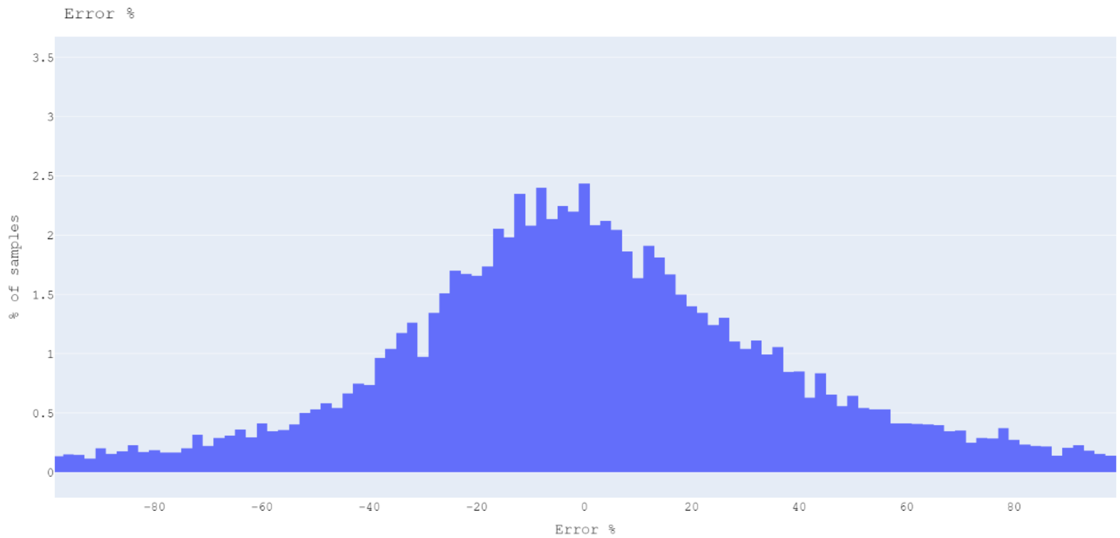


Figure 10. Percent error distribution in the ±100% range.
Note: Outliers excluded in this plot. More analysis of these performed later.

Further analysis of the error from the model indicates a pattern with the error over time. Before the 2000s the model didn't have many outliers relative to the amount of error within the range ±100% but as time increased the model was less successful at maintaining outliers. Part of it was accelerated during the financial crisis (2007-2008). See Figure 11.

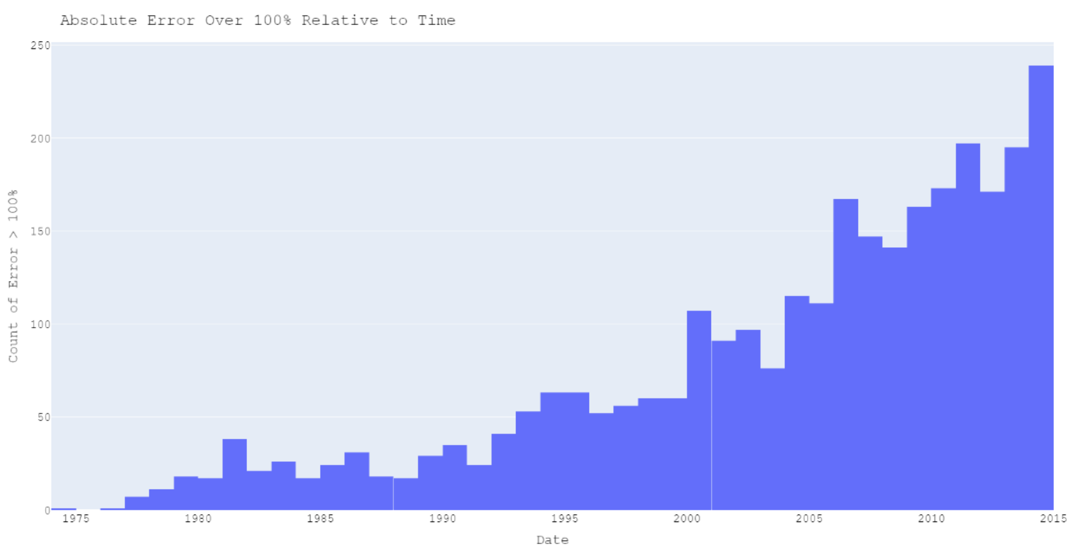


Figure 11. Error beyond ±100% over time.

By using this same graph and adding all the different sectors into it one can see the Financial Services sector became a majority contributor to the error along with the Technology sector (Figure 12). The Technology sector is not too surprising as companies in this sector typically have large evaluations (PE ratio large) relative to earnings which in most case are actually negative (Aigner & Schrabmair, 2020). If one wanted to better estimate for Technology companies one would have to compensate for earnings with customer growth and market growth.

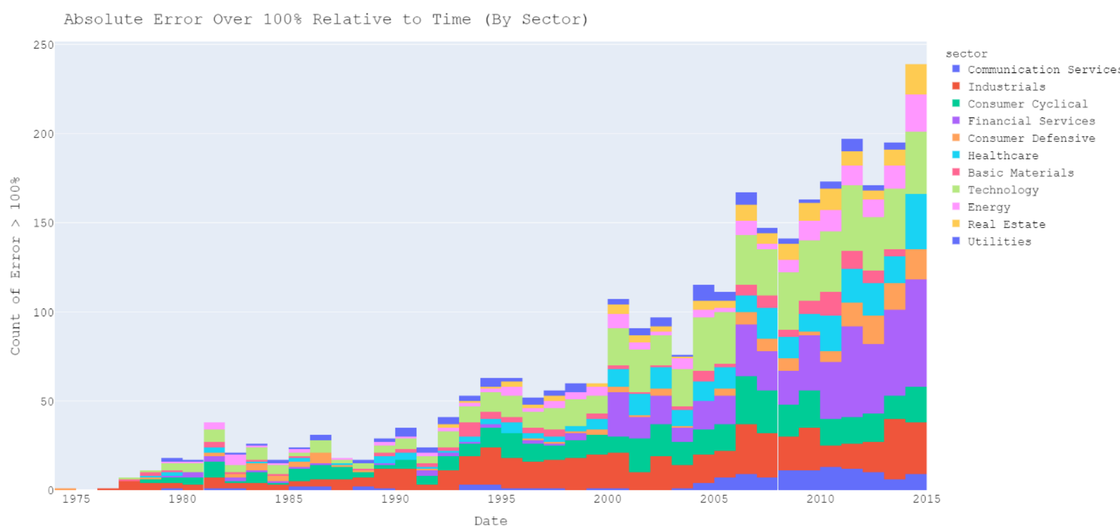


Figure 12. Error beyond ±100% over time by sector.

Another evaluation performed with this model was to choose a year and see how well it would perform if you were to buy the top 5 suggested stocks with a margin of safety of 50% relative to using Graham's formula. For Graham's method if you bought its top 5 suggested stocks and sold them in 5 years your average return would be 24% with some of them gaining as much as 121% and others losing as much as 58% (See Table 1 Below). This when compared to the S&P during the same time frame would have resulted in a gain of 33.38%.

Table 1. Top 5 Graham Suggestions.

Ticker	% return if sold in 5 years
KFS	121.26
SMSI	-58.99
RDCM	-0.28
SMBC	11.19
FMBI	46.75

The deep learning valuation model performed differently. It had selected a different group of 5 stocks from our test set. If you had bought the stocks it had suggested, you would have an average return of 67% after the 5 years which was better than the S&P during that time.

Table 2. Top 5 Model Suggestions.

Ticker	% return if sold in 5 years
IBOC	-13.00
FSM	163.09
ERII	294.99
PBT	-90.68
HCHC	-17.26

4.3. Evaluation to Linear Models

To further evaluate the Deep Learning model a comparison of its predictions to a common method used in Value Investing was performed. The most common approach observed in value investing is the linear growth method used by Phil Town in his book Rule 1 Investing. The math highlighted in (*8* explains how to calculate the intrinsic value using this linear growth model. In performing this analysis, only the test set of data was used. This test set was specifically reserved for testing and was never trained on. One additional change was that the discount from future value to present value of money is not applied to either model. This means comparison of only the future intrinsic value at a future time of 5 years, not the sum of discounted values between these times is used. This change required outputting only the terminal intrinsic value from the Deep Learning model to properly fit and align with the linear method. Note that since the discounts are constant for the linear model, the deep learning model, and the ground truth it can be ignored anyways.

The first analysis performed is of the Mean Squared Error (MSE) for each model when compared to ground truth. Shown in Figure 13 we observe that the Deep Learning model has higher counts at the lower MSE levels. This indicates a better fit with lower error when compared to the linear growth model. We can also see the counts increase for the linear model as the MSE increases. The MSE has been truncated at 1600, however it does continue well beyond this point to as much as 100,000 for the linear model and as much as 20,000 for the deep learning model. This means that the deep learning model has lower magnitude of outliers relative to the linear model.

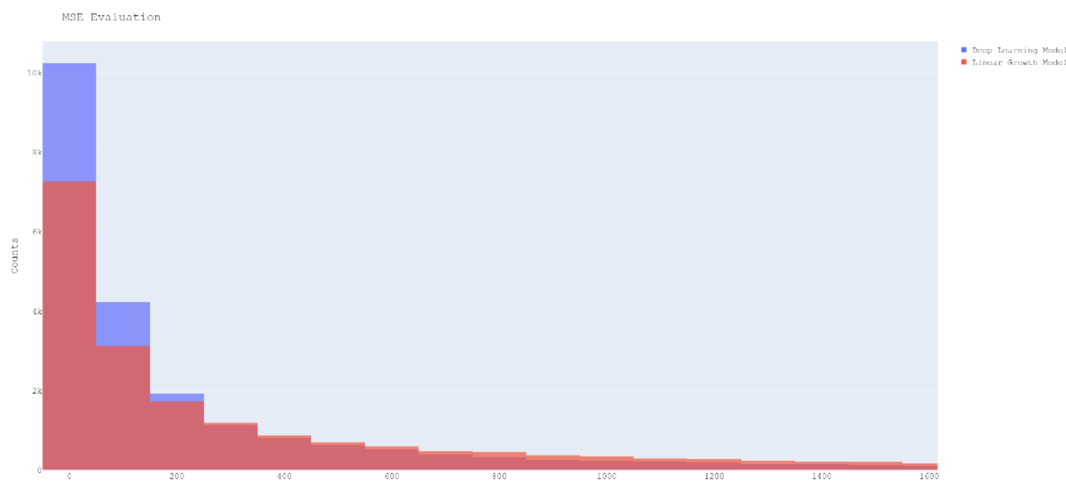


Figure 13. Mean Squared Error Comparison.

Moving from MSE to percent error of each model, we begin to see a similar pattern. Shown in Figure 14 is the percentage of error in the range -100 % to 100% for each model. We see that the linear model has lower counts within the range. This means it container a higher number of outliers outside the range and consequently is a worse fit for the underlying data. The centering of the error is also closer to 0 for the deep learning model indicating better normality across the residuals then a linear approach.

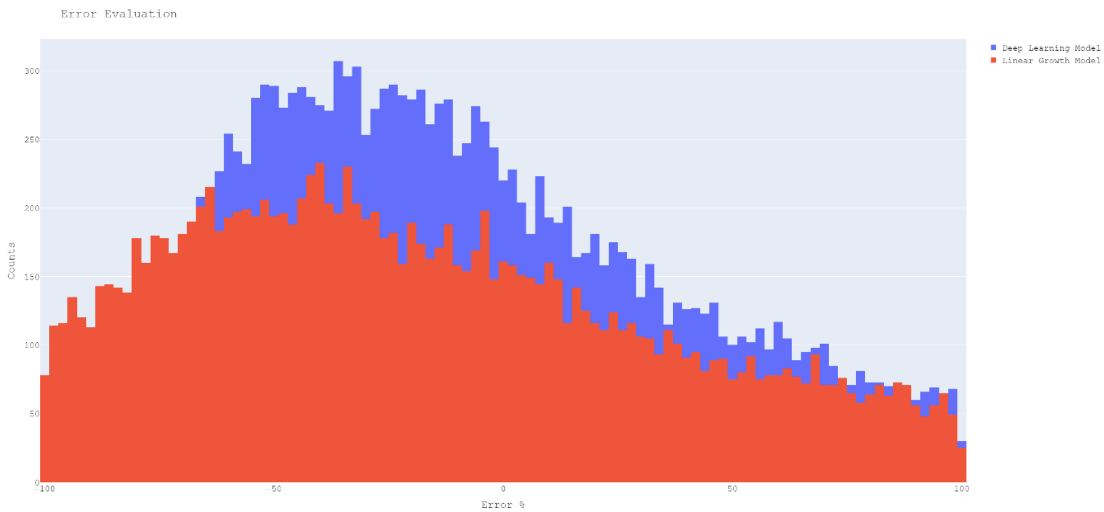


Figure 14. Percent Error Comparison.

When only looking at the outliers relative to time we arrive at Figure 15. This figure is a histogram showing the time on the x-axis along with counts of outliers on the y-axis. In observing the values highlighted in this chart, we can see that the deep learning model does better in the years when value investing became difficult as compared with the linear method. So even though there is still a pattern of outliers during this period it is substantially less than a direct linear growth model. Indications of this is that only using earnings and price to earnings ratio for valuation might be difficult in sectors where growth is more valued, like Technology as mentioned previously. In the discussions section we will discuss improvements to overcome this error, but it is worth highlighting here that it is an issue present in both models.

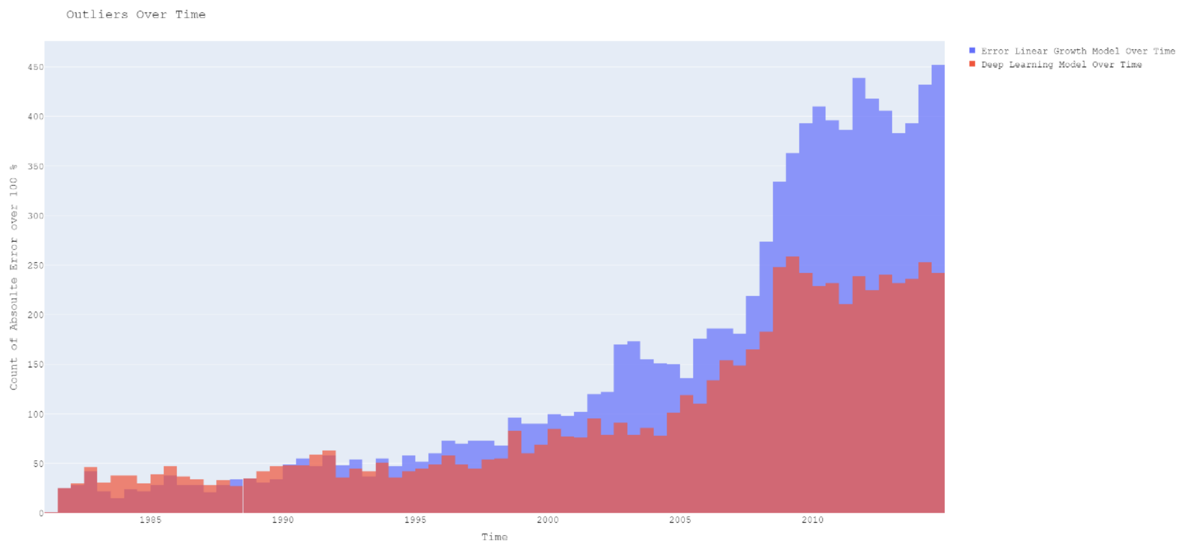


Figure 15. Error Comparison Over Time.

One last analysis that was performed was to look at the mean intrinsic value for ground truth, the linear model prediction and deep learning model prediction. It was performed over time to see any trends in error. Figure 16 captures this information. On the x-axis is time and on the y-axis are the mean intrinsic values for truth and the output for each model. By looking at this chart we can compare how well the mean intrinsic value during these periods were tracked for each model. On observing this chart, we see that the deep learning model better tracks the mean intrinsic values of companies when compared to the linear growth model. This implies that the Deep Learning model can better fit company valuations on average when compared to a linear growth model.

Figure 16. Performance Comparison Over Time.

In comparing the deep learning model to the linear growth model, we can confirm the initial hypothesis of our work regarding using Deep Learning for company valuation. That is that Deep Learning can better fit underlying dynamics of company valuation. Linear models fail to correctly capture the dynamics of company growth and reduction of value. Deep learning can better perform this analysis for the value investor providing more accurate estimates on future value. This is important as it is hard to look at every possible public company, a more automated approach to finding deals needs to exist which is the hope for our model.

5. DISCUSSION

5.1. Example Usage

After talking through the model performance and implementation it is worth mentioning how one would directly use these models. Let's say we are interested in finding undervalued stocks in the sector of Technology. We first gather 5 years of historical fundamental information for all the companies in the Technology sector. We then scale this data appropriately into data the model can interpret (standard normalization technique, enterprise value, min/max scaling). We run each of these companies through the model and end up with a net present value for each. This net present value considers all future cash flows predicted and discounted back to the present. If we then compare this price to the current market price we arrive at a buy or sell decision. We would want to buy companies whose predicted present value is above the current market price by an amount called the margin of safety. Similarly, if we own any existing companies, we will want to look at if they are overvalued. A company is overvalued if its current market price multiplied by the margin of safety is higher than the predicted present value from the model; see Figure 13 for an illustration of using the model.

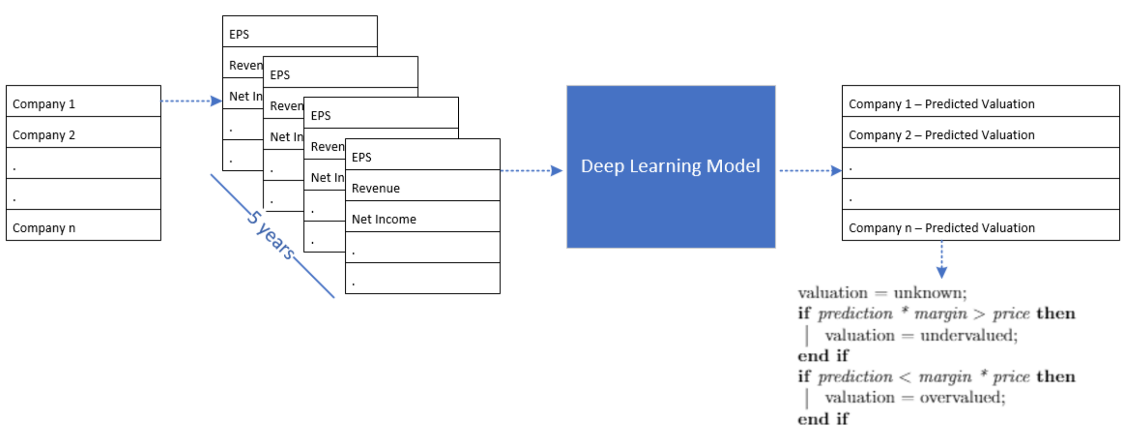


Figure 17. Model Usage

One major benefit of this model is that it can be ran frequently to find deals for the investor by automatically filtering out companies that are not undervalued. This can save a great deal of time and effort for investors. There are stock screeners that do Grahams formula comparison as well as linear approaches, however, as was demonstrated in the results section the deep learning model performs better and can uncover more undervalued companies for the investor.

5.2. Recommended Improvements

The models presented in this paper were found to perform very well on the test set of data reserved for evaluation. However, there are areas for improvements in both cases. As was seen for the valuation model, the present market conditions don't favor earnings relative to what they have in the past. A better valuation metric might be predicting actual sector growth, customer acquisitions and revenue growth. Another gap is that using only EPS and PE misses other company cash flows like dividends. Dividends are positive return to the investor and should be factored into the future cash flows along with earnings. The largest gap though is not considering macro effects such as Gross Domestic Product (GDP), Consumer Price Index (CPI), sector capitalization and inflation. The 2007 financial crisis and the 2020 coronavirus pandemic both have long term effects on the market and company performance. Adding macro effects would help the model predict economic impacts early on which would better reflect future company outlook and cash flows. An added benefit of including macro effects would be less variability in the model predictions leading to better results in training and evaluation.

Other improvements that could be looked at are related to Deep Learning itself. There are promising results with different training techniques and network architectures. For example, Generative Adversarial Networks (GANs) are shown to better model datasets (Goodfellow, 2016). Transformers are showing great promise in replacing LSTMs for time series data (Wu, Green, Ben, & O'Banion, 2020). Using these techniques could help the models learn more from the existing dataset and perform better on test sets.

5.3. Current Applications

Although the models could be improved using more data and different modeling techniques, they already show promise in helping the intelligent investor make decisions. Applying the models to use in a filtering tool would be extremely useful for investors. Currently it is a full-time job for investors to choose companies due to the large amount of publicly traded companies. If only half of them could be filtered out that could be extremely useful and save a ton of time. The models could currently do this with a good degree of accuracy (20% outlier rate).

Another direct application of the existing models is usage as an algorithmic mutual fund. Mutual funds allow for a high level of capital due to multiple investors mutually allocating their investments into the fund. Allocating this capital in the top choices from the model could potentially achieve better performance than the S&P as outlined in example in table 2.
Lastly, the best usage for these models is in conjunction with other core tools value investors use. Does the company have a good moat? Are the managers views in line with your own? How does the model estimate future cash flows? Answering questions like these helps make a type of decision tree that can lead to better decisions then just using one decision alone. i.e. just cause a company has a good moat if the managers are not great then they could make bad decisions leading to the company failing. Or just because the model says a company has a high valuation relative to its price it could still fail due to not having a moat and being replaced with a

competitor. Therefore, it is important to use these models in conjunction with other value investing tools. This will allow risk to be mitigated and provide better company outlooks.

REFERENCES

Aigner, D. A., & Schrabmair, W. (2020). Graham's Formula for Valuing Growth Stocks. *from arXiv:2003.09300 (available at https://arxiv.org/ftp/arxiv/papers/2003/2003.09300.pdf)*.

Alberg, J., & Seckler, M. (2021, April 1). *deep-learning-long-term-investing*. Retrieved from euclidean: https://www.euclidean.com/deep-learning-long-term-investing-1

Bishop, C. M. (2006). *Pattern recognition and machine learning.* New York City: Springer.

Buffett, W. &., & Fund, E. G. (2014). *Berkshire Hathaway Letters to Shareholders.* Palo Alto California: Max Olsen.

Clayton, A. (2020). *Deep-Learning the Cash Flow Model.* New York City, New York: Moody's Analytics.

Elend, L., Tideman, S. A., Lopatta, K., & Kramer, O. (2020). Earnings Prediction with Deep Learning. *Advances in Artificial Intelligence*, 267–274.

Fridman, L. (2021, February 7). *Richard Craib: WallStreetBets, Numerai, and the Future of Stock Trading* |. Retrieved from lexfridman.com: https://lexfridman.com/podcast/

Goodfellow, I. B. (2016). *Deep Learning.* London: MIT Press.

Graham, B. (2003). *The Intelligent Investor Rev Ed* (Revised. ed.). New York City, New York: Harper Paperbacks.

Graham, B., & Dodd, D. (1996). *Security Analysis.* New York City, New York: McGraw-Hill.

Lutz Kruschwitz, A. L. (2020). *Stochastic Discounted Cash Flow: A Theory of the Valuation of Firms.* New York City, New York: Springer.

Marshall, K. J. (2017). *Good Stocks Cheap: Value Investing with Confidence for a Lifetime of Stock Market Outperformance.* Columbus, OH: McGraw-Hill Education.

oinkina. (2015, 08). *Understanding-LSTMs.* Retrieved from github: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Pollan, M. (2002). *The Botany of Desire: A Plant's-Eye View of the World.* New York City, New York: Random House.

Prado, M. L. (2018). *Advances in Financial Machine Learning.* Hoboken, New Jersey: Wiley.

Schroeder, A. (2008). *The snowball : Warren Buffett and the business of life.* New York City, New York: Bantam Book.

Sepp Hochreiter, J. S. (1997). Long Short-Term Memory. *Neural Computation*, 1735-1780.

Town, P. (2007). *Rule #1: The simple strategy for successful investing in only 15 minutes a week!* New York City, New York: Random House.

Town, P. (2021, March 26). *ruleoneinvesting*. Retrieved from investment-calculator: https://www.ruleoneinvesting.com/investment-calculators/

Wu, N., Green, B., Ben, X., & O'Banion, S. (2020). Deep Transformer Models for Time Series Forecasting:- The Influenza Prevalence Case. *arXiv:2001.08317 (available at https://arxiv.org/abs/2001.08317)*.

APPENDIX. DEFINITIONS OF VARIABLES

What follows are a definition of the variables used in the deep learning model as well as how to derive features that were engineered based off of (Marshall, 2017).

Table A1. Data source Variable Names/Definitions.

Variable name	Location	Explanation
capxy	Compustat	Capital Expenditures
dpq	Compustat	Depreciation and Amortization – Total
ibq	Compustat	Income Before Extraordinary Items
dlttq	Compustat	Long-Term Debt - Total
cshfdq	Compustat	Com Shares for Diluted EPS
oibdpq	Compustat	OIBDPQ -- Operating Income Before Depreciation - Quarterly
cshprq	Compustat	CSHPRQ -- Common Shares Used to Calculate Earnings Per Share
adjex	Compustat	ADJEX -- Cumulative Adjustment Factor by Ex-Date
cheq	Compustat	Cash and Short-Term Investments
rectq	Compustat	Receivables - Total
invtq	Compustat	Inventories - Total
ppegtq	Compustat	Property, Plant and Equipment – Total (Gross) - Quarterly
actq	Compustat	Current Assets, Total
dlcq	Compustat	Debt in Current Liabilities
uaptq	Compustat	Accounts Payable - Utility
lctq	Compustat	Current Liabilities - Total
lltq	Compustat	Long-Term Liabilities (Total)
revtq	Compustat	Revenue Total
cogsq	Compustat	Cost of Goods Sold
xsgaq	Compustat	Selling, General and Administrative Expenses
req	Compustat	Retained Earnings
niq	Compustat	Net Income (Loss)
prccq	Compustat	price close
prchq	Compustat	price high
prclq	Compustat	price low
epsfxq	Compustat	Earnings Per Share (Diluted) – Excluding Extraordinary items
epspxq	Compustat	Earnings Per Share (Basic) – Excluding Extraordinary items

Table A1. Data source Variable Names/Definitions (continued).

Variable name	Location	Explanation
Icaptq	Compustat	ICAPTQ -- Invested Capital - Total - Quarterly
cshoq	Compustat	Common Shares Outstanding
working_capital	Derived	Assets - Liabilities
market_cap	Derived	Shares Outstanding * price
total_debt	Derived	Long term + Short Term Liabilities
book_value	Derived	Total assets−Total liabilities
enterprise_value	Derived	market capitalization + total debt - cash
leveraged_free_cash_flow	Derived	Net Income + Depreciation + Amortization – Change in net working capital – capital expenditures – mandatory debt payments
return_on_captial_employed	Derived	Operating income / capital employed
free_cash_flow_return_on_capital_employed	Derived	leveraged free cash flow / capital employed
liabilities_to_equity_ratio	Derived	Total debt / book value
pe_ratio	Derived	Price / earnings per share
sector	Yahoo	Sector (Technology, IT, etc.)
industry	Yahoo	Industry (Consumer Electronics, Consumer Services, etc.)