Dividend Champion Predictor

Connor DePalma
ECE Dept.
Stevens Institute of Technology
Somerville MA, United States
cdepalma@stevens.edu

Tim Demetriades
ECE Dept.
Stevens Institute of Technology
Staten Island NY, United States
edemetri@stevens.edu

Dan Pelis
ECE Dept.
Stevens Institute of Technology
Hoboken NJ, United States
dpelis@stevens.edu

***Abstract*—Predicting the performance of a stock is not an easy task. It is something people have spent years working on, with countless machine learning models having been created to try to assess whether a certain stock is a good or bad investment. Dividends, which are payments made to its shareholders, can be used to determine how well a company is performing. Dividend champions are companies that have been paying out increasing dividends for over 25 years. This makes these companies worthy investments, while at the same time if a company loses its status as a Dividend Champion it may be a good idea to avoid investing. As there are currently no machine learning models that specifically focus on Dividend Champions, we proposed to create a model that can predict whether a company will lose its status as a Dividend Champion based on historical metrics of current and past Dividend Champions to determine if a company is worth investing in. Since the data was time series data, an RNN with LSTM cells was used to create the model. Preliminary results for the model were promising, with the model doing a relatively good job at predicting future dividend values and therefore predicting if a company will lose its champion status.**

I. Introduction

Dividend Champions are companies that have kept a streak of increasing their dividends for a minimum of 25 years. These stocks are held in high regard to those who practice the investing strategy known as ”Dividend Snowball”, which relies on the compounding effect caused by reinvesting dividends. As a result, it is generally smart and less risky to invest in Dividend Champions as they have a proven track record of performing well even during times of recession. The purpose of titles such as ”Champion” indicates to investors that these companies have and will continue to provide a reliable stream of dividends for many years. Therefore, we propose a system that will analyze the historical data of the current and past Dividend Champions to predict whether a company will lose its titles in the near future.

The Dividend Investing Resource Center (dripinvesting.org) is a centralized hub for research and information about dividend investing. The source provides monthly reports including the list of the current Dividend Champions and many metrics such as their share price and dividend yield. These reports can be used to create a reliable list of Dividend Champions dating back to 2010, including the companies which held the Champion title but later lost their status. From this list the Yahoo Finance API can be called and accurate quarterly data can be queried and organized. The prepossessing steps above result in a robust dataset of past and present Dividend Champions along with their metrics, which can be then be used in a variety of regression or classification algorithms.

The dataset described above allows itself to be used for either classification or regression, however since the features consist of time series data it is intuitive to perform regression on the set. More specifically a common solution for time series regression is Recurrent Neural Networks (RNNs). RNNs are great at modeling sequential data due to their looping architecture that allows signals to travel in both direction, essentially providing memory to the network. However, a simple RNN is usually not particularly accurate due to the common issue of vanishing gradients. The vanishing gradient problem causes these RNNs to have short-term memory, meaning they have trouble remembering earlier data and as a result will makes its predictions mostly from the most recent data. There is a solution though, in the form of LSTMs and GRUs. The Long Short Term Network (LSTM) and Gated Recurrent Unit (GRU) variations of the classic RNN algorithm will help avoid the problem of vanishing gradients and help improve results.

The advantage of using an LSTM over a regular RNN comes in the form of the long-term cells. These cells are capable of learning to store important information while disregarding unimportant information. They then can keep this information for as long as needed where it can then be extracted. This naturally provides a form of long-term memory to the model, making them superior to regular RNNs and the architecture of choice when it comes to time-series data.

As for the preliminary results, the team plotted the prediction of the network with the true values the test samples to visually determine the performance of the model. It can be seen that for the companies that lost champion status, the model does a pretty good job at predicting the following dividend value given the previous two dividend values. The mean squared error (MSE) was also calculated and was relatively low. After making an adjustment for lag, the model produced even better results. These results made it easy to determine if a company was to lose its status as a Dividend Champion.

Many of the solutions to similar problems the team found relating to stock forecasting use various models, such as multiple linear regression, support vector regression, and neural networks. However, the majority of these models focus on predicting the future price of stocks. While this may be useful in determining if a stock is worth investing in, we believe that focusing on Dividend Champions can potentially be a superior determiner in whether to invest in a stock. Additionally, the use of RNNs with LSTM cells allows for long-term memory, meaning the model is able to use not just recent metrics but a long sequence of metrics for its forecasting, resulting in higher accuracy.

II. Related Work

Being that Dividend Champions are generally well-established and profitable companies, they are generally less of a risk to invest in and therefore desirable to many people. This is why people started tracking Dividend Champions, such as Dave Fish who started the spreadsheets that contain a list of all the Dividend Champions along with Contenders (companies who have increased their dividend for the past 10 years) and Challengers (5 years) and is the source of our dataset.

Although these Dividend Champions have been tracked for over 10 years, from what we have found there are no machine learning models that have been created to make predictions based on these lists. While many machine learning models have been created to forecast stock prices and predict whether or not to invest in a stock based on such metrics, there are currently no models that are able to predict whether or not a Dividend Champion is going to lose its status, which can be an important metric in determining whether or not to invest in a stock.

Many of the related work found focuses on predicting the future price of a stock. For instance, David Enke, Manfred Grauer, and Nijat Mehdiyev introduced a way to use multiple regression and type-2 Fuzzy Clustering to create prediction models for forecasting stock market prices [1]. Multiple regression is similar to regular linear regression, although instead of there only being one independent variable there are several (different metrics can influence the price of a stock), while Type-2 Fuzzy Clustering uses type-2 fuzzy sets to make predictions. In the paper, the three explain how they created a hybrid of the models to predict stock price levels.

Support Vector Regression (SVR) is another machine learning technique that can be used for regression. SVR is similar to the popular technique of Supper Vector Machines (SVM), although instead of trying to minimize the number of points within the decision boundaries, SVR tries to maximize this. Bruno Henrique, Vinicius Sobeiro, and Herbert Kimura proposed the use of SVR models to predict stock prices for large and small capitalisations and in three different markets [2].

The use of neural networks is also very common with both regression and classification tasks. Neural networks are made up of multiple nodes and layers, with each node behaving similarly to multiple linear regression, and the network tries to find correlations between many different inputs. However, unlike with multiple regression, neural networks can handle non-linear data as a result of the use of non-linear activation functions. Donglin Chen and Dissanayaka Seneviratna attempted to use feed forward back propagation neural networks (BPNN) to conduct one step ahead forecasting on stock prices indices [3].

A common trend among all of these solutions that were found is that they all focus on forecasting stock prices. While this is certainly a good way to predict whether a stock is worthy investment, our solution introduces the idea of Dividend Champions, focusing specifically on metrics such as the dividend yield to predict whether or not a company will lose its champion status, which can potentially be a better determiner if a certain stock is a good or bad investment.

III. Our Solution

A. *Description of Dataset*

Being that there is no publicly available dataset containing an organized list of Dividend Champions and their historical data, a dataset needed to be created for this experiment. Originally the dataset was to be comprised of Dividend Aristocrats, which are stocks with a 25 year streak of increasing dividends while also being part of the S&P 500 Stock Index. However being that these conditions make the Aristocrat list quite short (around 65), it was more efficient to study the Champions list which provides many additional companies for training, as the champions are not required to be part of the S&P 500.

The Dividend Investing Resource Center is an online resource that provides **monthly** reports on the current Dividend Champions, as well as some other Dividend related groupings. These reports contain the list of the current companies maintaining the title of "Dividend Champion" and the stock metrics (such as stock price and dividend yield) at the time of the report. Due to format inconsistency and incomplete data, we found the best time period to use these reports is from 2010-2020. From these reports two lists can be compiled, one containing the current dividend champions and one tracking companies who were Champions at some point within the time period, but lost their status. Without this resource the team would have needed to iterate through a multitude of data (likely through a financing API) to find companies that maintained increasing dividends for 25 years only to lower payouts in a subsequent year in order to find the companies that lost their status as Dividend Champions.

An example of a company, Washington Real Estate Investment Trust, that lost status can be seen in Figure 1 below. As shown, this company had increasing dividends for several years, but then had a significant reduction in their dividend payout for a sustained period of time starting around July 2012 for long enough for there to be an overall negative change from the previous year. This is exactly what constitutes a loss of Champion status.

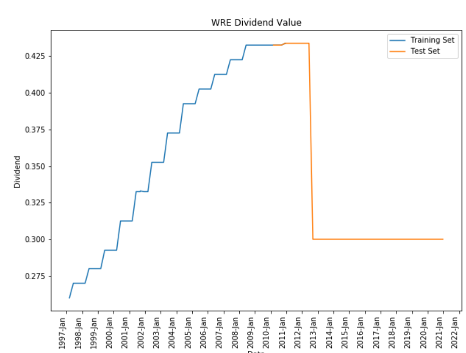

Fig. 1: Lost Status

The lists were separated to enable to dataset to be used for either regression or classification implementations. These lists of companies were then used to collect accurate historical data through the Yahoo Finance API. This API provides robust historical data on a given stock and offers the ability to collect various stock metrics. For a regression implementation the distributed dividends of the companies were collected and processed.

Being that dividend distributions are up to the discretion of the company this metric can be quite inconsistent. For instance the company could choose to not pay dividends for a given period, which is usually uncommon for Dividend Champions since they are generally stable and successful companies and since it would cause them to lose their status as Dividend Champions. However, it does sometimes occur, and this introduces missing values into the dataset. To combat this zeros need to be added in these missing periods, since zero is an accurate representation of the dividend amount distributed for that period.

A source of anomalies is a type of dividend called *Special Dividends*. These are unscheduled one-time dividends which are usually larger than the normal dividend stream a company puts out. In terms of the dataset, these special dividends would produce spikes that act as noise and would hurt a models ability to find a trend in the regular dividend stream, therefore they need to be removed. Figure 2 contains a visual representation of the dividend values for SJW Group, which contain a noisy sample around January 2018 (in this case a sharp decrease followed by a sharp increase). This sample should not automatically remove it's Champion status, as it was likely an adjustment or additional dividend provided by SJW Group to its shareholders. By decreasing the impact of the noisy Special Dividends, the team is ensuring that the generated models will not accidentally predict the loss of Champion status.

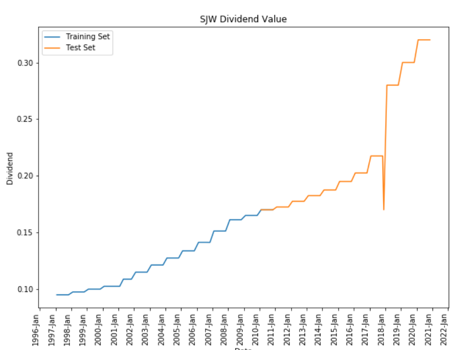

Fig. 2: Kept Status With Noise

Finally, dividends do not have to be paid on a quarterly basis. It is common to find companies that pay their dividends on a monthly basis, meaning these dividends need to be combined into a quarterly yield to be properly compared with the other companies.

B. *Machine Learning Algorithms*

Recurrent Neural Nets (RNNs) use connections along a temporal sequence to process variable length input sequences. These RNNs keep an internal hidden state, often referred to as memory, to keep track of the information they have already seen. These neural networks create recurrent connections by using the output of the preceding step as the input for the current step. Given the temporal nature of RNNs, they are a common solution applied to time series analysis problems. A limitation of RNNs comes when the model is applied to a long time series (usually hundreds of time steps). This long input sequence translates to a large amount of steps in the RNN, which becomes a problem during gradient descent since the algorithm must propagate back through these steps. This causes the recurrent weight that connects the layers to become very small leading to the values sent through the network to get smaller with each step. This issue is known as the **Vanishing Gradient** problem.

Due to the transformations the data goes through when traversing an RNN, some information is lost at each time step, which causes the RNN state to essentially forget the beginning of the sequence. Long Short-Term Memory (LSTM) networks, a variant of RNNs, address this and the vanishing gradient with the use of long-term memory, known as *cell state*. An LSTM cell is similar to a regular cell except that it splits the state into two vectors - the hidden (short-term) state and the cell (long-term) state. In doing so, it allows the network to learn what to remember and store in the cell state and what to forget. The LSTM can then preserve this for as long as it is needed (and forget it when it's not needed) and extract it when it's needed. As a result, while LSTMs may be more complex than regular RNNs, they are able to remember much more information, producing superior performance and thus have made regular RNNs almost irrelevant.

The initial design of the teams' LSTM model consisted of an arbitrary number of LSTM layers and hidden units, which were later tuned after experimentation. The Dropout layers were added between the LSTM layers to statistically mask weights and reduce overfitting throughout the network. Since LSTMs were used there was no need to choose activation functions. LSTMs by definition use a tanh activate function for cell state and a sigmoid activation function for cell output. The model ends with a single Dense layer with an output of size one, representing the stocks dividend yield for the succeeding time period. The current design after tuning can be view in 3.

C. *Implementation Details*

As mentioned in the previous section, the team currently has an LSTM model 3 as the first candidate algorithm that could provide a solution to the proposed problem. Similar to the dataset creation and clean up process, the model was written in Python3 using Jupyter Notebooks. This environment allowed for easy testing and debugging of the model by allowing for inline visualizations and intermediary outputs. The model was written using Google's Keras library, which contains high-level classes and methods that wrap TensorFlow (Googles core AI framework) functions, objects, etc. The use of Keras allowed for the team to quickly test different LSTM architectures and tune hyper-parameters.

The hyper-parameters were tuned through validation testing of the model. This involved using a subset of the training samples (*not* testing) as validation for the accuracy of the model at each epoch, known as the validation set. Since a

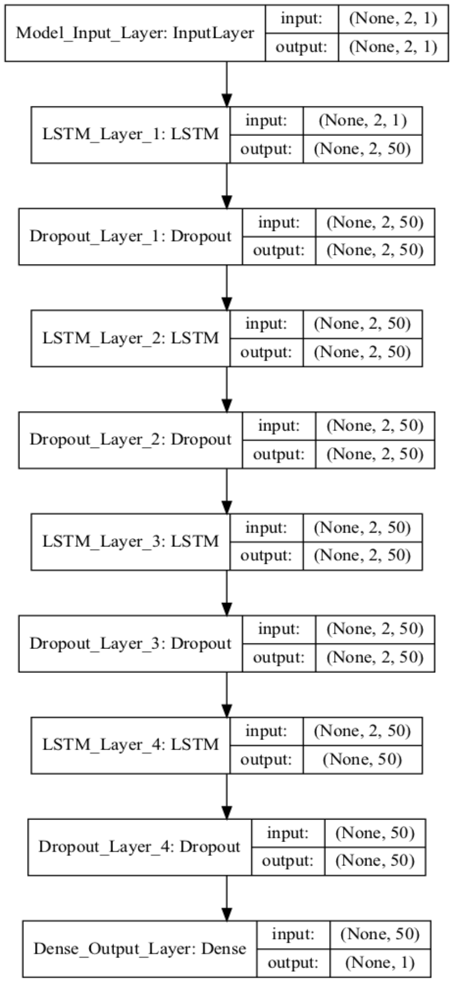

Fig. 3: LSTM Model

regressive model was chosen, ideally the predictions would align closely with the testing samples themselves. This meant that a simple loss function like Mean-Squared-Error (MSE) would suffice at least for this initial implementation. The validated results were based off of the MSE measurements and helped the team decide whether a parameter change helped or hindered the model.

The teams preliminary results for the LSTM algorithm implementation are visually based. Since the intention of the model was to ingest a handful of samples and predict the next sample, the best way to test the performance of the model is to simply plot the prediction along side the true values of the test samples. The Washington Real Estate Investing Trust, the same company shown in Figure 1, was pulled out as an example since it was known that the company lost its status. In Figure 4 it can been seen that the model does a decent job of predicting the next dividend value given two previous dividends.

This resulted in a relatively low MSE, however there is lag introduced by the LSTM based model that would need to be compensated for in order to provide more accurate predictions. An example of the same results for the Washington Real Estate Investing Trust are shown in Figure 5 but with a lag

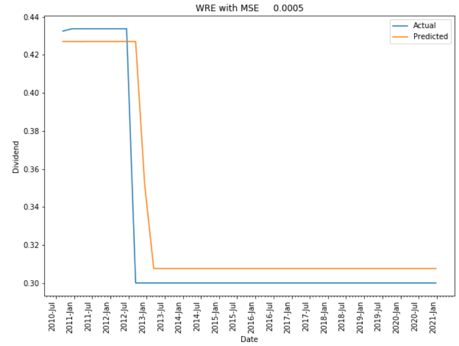

Fig. 4: Model Prediction Example

adjustment of a single time step. It is can easily be seen how this improves the model prediction, to the point where the team could possibly use these results to quickly determine that this company would be on the verge of losing status around July 2012.

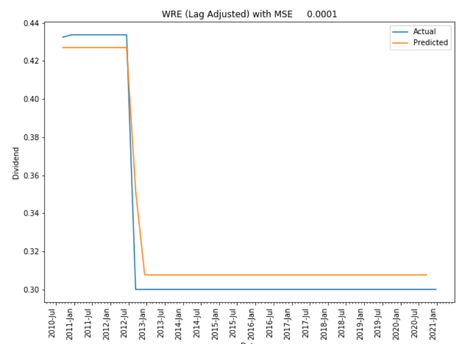

Fig. 5: Model Prediction Example With Lag Adjustment

These results place the team in a good position to be able to provide a meaningful solution based on multiple Machine Learning models by the end of this project.

References

[1] David Enke, Manfred Grauer, Nijat Mehdiyev, Stock Market Prediction with Multiple Regression, Fuzzy Type-2 Clustering and Neural Networks, Procedia Computer Science, Volume 6, 2011, Pages 201-206, ISSN 1877-0509, https://www.sciencedirect.com/science/article/pii/S1877050911005035

[2] Bruno Miranda Henrique, Vinicius Amorim Sobreiro, Herbert Kimura, Stock price prediction using support vector regression on daily and up to the minute prices, The Journal of Finance and Data Science, Volume 4, Issue 3, 2018, Pages 183-201, ISSN 2405-9188, https://www.sciencedirect.com/science/article/pii/S2405918818300060

[3] Chen, D. and Seneviratna, D. (2014) Using Feed Forward BPNN for Forecasting All Share Price Index. Journal of Data Analysis and Information Processing, 2, 87-94. doi: https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/journal/paperinformation.aspx?paperid=51486