SAMU BREITE

Three-factor model in deep neural networks

Master of Science thesis
Faculty of Management and Business (MAB)
Examiner: Prof. Juho Kanniainen
Examiner: Prof. Moncef Gabbouj
June 2021

Abstract

SAMU BREITE: Three-factor model in deep neural networks
Master of Science thesis
Tampere University
Master's Degree Programme in Industrial Engineering and Management
June 2021

Empirical asset pricing literature has widely recognised different factors and they are a well researched topic. The main approach in past research regarding factors in empirical asset pricing has been based on linear models. Few studies have considered neural networks in the context of empirical asset pricing and factors. Neural networks have the potential to solve some deficiencies that the traditional linear approach has and they have shown their potential in many fields. This thesis considers the size, value and momentum factor model with neural networks with the aim of showing neural networks' potential in empirical asset pricing and the aim of learning about their interpretation of factors.

Firstly, the thesis considers a regression prediction task for US stocks. The thesis uses anchored walk forward training and testing. The thesis shows that a multilayer perceptron neural network has prediction power and predicts returns better than a zero return forecast. The thesis also forms comparable traditional linear factor models and compares their prediction capability to neural networks' prediction capability. A multilayer perceptron has better prediction capability than traditional linear models when root mean squared error and R^2 statistic are considered.

Secondly, the thesis considers a classification prediction task for the same stock returns. Again, anchored walk forward training and testing process is used. The neural network used in this task is a temporal attention bilinear neural network. It is shown in this thesis that it can be more accurate at predicting stock returns, measured with F-Score, than logistic regression which is a comparable linear model. The thesis also investigates attention patterns from the temporal attention bilinear networks. These patterns reveal a neural network focusing on different instances to distinguish riskier stocks from less riskier and focus on same instances when differentiating if the riskier stock performs well or badly. Attention patterns also suggest that all factors considered in the thesis are relevant. Lastly, the thesis formulates investment strategies based on predictions by neural networks. These strategies show that predictions have economic meaning and investors can gain from them.

Keywords: Neural Network, Empirical Asset Pricing, Size, Value, Momentum

Tiivistelmä

SAMU BREITE: 3 faktorin malli neuroverkoilla
Diplomityö
Tampereen yliopsito
Tuotantotalouden tutkinto-ohjelma
Kesäkuu 2021

Faktorit ovat laajasti tunnistettu ilmiö kirjallisuudessa. Yleinen lähestymistapa faktoreihin on ollut muodostaa niistä lineaarinen hinnoittelumalli. Muutamat tutkimukset ovat myös hyödyntäneet neuroverkkoja faktoripohjaisissa hinnoittelumalleissa. Neuroverkot ovat osoittaneet hyödyllisyytensä monissa eri yhteyksissä. Neuroverkoilla on ominaisuuksia, jotka saattavat korjata lineaaristen hinnoittelumallien aiemmin tunnistettuja ongelmia. Tämä tutkimus pyrkii osoittamaan neuroverkkojen potentiaalin hinnoittelumalleissa ja tarkastelee neuroverkkojen tulkintaa faktoreista. Tätä varten tutkimus muodostaa faktorimallin neuroverkoilla, joka koostuu arvo-, koko- ja momentum- faktoreista. Aineistona tutkimuksessa käytetään Yhdysvaltojen osakemarkkinoiden dataa.

Ensin tutkimus keskittyy regressioennustamaan osakkeiden tuottoa käyttäen ankkuroitua eteenpäin kävelymetodia opettamiseen ja testaamiseen. Tutkimus näyttää, että monikerroksinen perseptronineuroverkko ennustaa tuottoja R^2-luvulla mitattuna paremmin kuin nollatuotto-odotus. Näiden neuroverkkojen ennusteita verrataan myös lineaarisen hinnoittelumallin tuottamiin ennusteisiin. Tuloksista huomataan että, perseptronineuroverkko tuottaa lineaarista mallia parempia ennusteita virheen neliöllisellä keskiarvolla ja R^2-luvulla mitattuna.

Toiseksi tutkimus keskittyy luokitteluennustamiseen hyödyntäen taas ankkuroitua eteenpäin kävelymetodia. Luokitteluun käytetään ajallisen huomion bilineearista neuroverkkoa ja logistista regressiota. Neuroverkko voi olla tässä yhteydessä parempi ennustamaan osakkeiden tuottoja F-pisteillä mitattuna kuin logistinen regressio. Neuroverkon huomioiden perusteella tutkimus toteaa neuroverkkojen erottelevan riskisemmät osakkeet vähemmän riskisistä keskittymällä tiettyihin instansseihin. Huomioiden perusteella voidaan myös todeta neuroverkon keskittyvän samoihin tekijöihin hyvien ja huonojen osakkeiden kohdalla. Huomiot myös viittaavat siihen, että kaikki käytetyt faktorit ovat relevantteja. Tämän lisäksi tutkimus muodostaa sijoitusstrategioita neuroverkkojen ennusteiden pohjalta. Strategiat osoittavat, että sijoittajat voivat hyötyä neuroverkkojen ennusteista.

Avainsanat: Neuroverkot, Faktori hinnoittelu, Koko, Momentum, Arvo

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

Preface

The topic that I chose for this thesis was not the easiest one. At times, the task of completing this thesis has felt quite hard. Nevertheless, the topic has always been interesting and the interest about this topic was the reason why initially started with it. My interest towards factors and machine learning has only deepened during the thesis. I have learned so much about neural networks, implementation of neural networks and machine learning during the process of writing this thesis. Also my knowledge about factors and empirical asset pricing has deepened as a result of this thesis. Hopefully, reader of this thesis can learn about these topics too as I have. I believe, that I can benefit from the lessons I have learned during the writing of the thesis as I continue to work with similar topics. In the end, I am glad about how I chose to do my thesis and I am glad that it is done.

I like to thank couple of different people that have helped me to accomplish this thesis. First, I like thank my examiners Prof. Juho Kanniainen and Moncef Gabbouj for providing me helpful and vital feedback. I also like to thank my family for support during the writing process. I have also found many conversations with my colleagues very helpful at providing perspective to the topic.

Contents

List of Figures

List of Tables

List of abbreviations and symbols

BE	Book equity
CAPM	Capital asset pricing model
HML	High minus low
ME	Market equity
MLP	Multilayer perceptron
OLS	Ordinary least squares
R_m	Market risk premium
RMSE	Root mean squared error
SMB	Small minus big
TABL	Temporal attention-augmented bilinear

1 Introduction

In recent years, factor investing and quantitative equity strategies have become more and more popular. There are vast amounts of funds and exchange traded funds (ETFs) that follow factor-based strategies. Today it is easy to pick and choose what factors and factor-based strategies you as an investor want in your portfolio. Today, a large portion of money invested in equity markets follows systematic factor-based strategies or indexes. Indexes, in many cases, can be seen as a type of systematic strategy that usually considers firm size. Factor-based strategies have proven their ability to produce out-performance over time. However, factors can also have time periods when they under-perform. Given all of this, factors are a meaningful part of today's equity markets. Many pension funds, insurers and endowments incorporate factors in their investment strategies. This further highlights the importance of factors as these institutions affect on many people's lives. Factors are widely considered phenomena in equity markets and not just a fringe investment strategy considered by few. Understanding factors' relation to returns better is still important given their role in today's equity markets and investing. Better understanding of factor returns can lead to higher returns and better diversification of risks. More precise return predictions have value themselves as they are useful for economic research.

Many characteristics of stocks have been found to be linked to the stocks' returns. Currently, the literature recognises over 300 characteristics that are linked to stocks' returns (Harvey et al. 2016). The literature has found evidence in different markets and asset classes that characteristics are linked to the returns. There is a long tradition of forming factor-based asset pricing models, thus predicting returns based on characteristics. Traditionally, models have been based on sorted portfolios and linear regression. Traditional models have some limitations when it comes to predicting returns. They do not capture nonlinear behaviour and they have trouble handling a large number of predictor factors. Characteristics are also often correlated and dependent on each other, making their relation to returns more complex.

Machine learning models solve some of the problems that traditional models face in empirical asset pricing. Machine learning can help in asset pricing by allowing the usage of advanced statistical models and nonlinearity. Applying machine learning to return prediction may provide a better forecasting result and help to better understand returns. Machine learning has evolved a lot in recent years, and it has been used to solve many problems in many different fields. Some studies have applied machine learning to finance problems. Recent studies of Gu et al. (2018) and Feng et al. (2018) have found promising results when it comes to applying machine learning to empirical asset pricing and recognising its potential.

Neural networks are widely considered to be the most promising and advanced machine learning model in many fields. They are a deep learning black-box method. Neural networks have been able to solve many problems in different fields that have been considered very difficult or even impossible. Gu et al. (2018) and Feng et al. (2018) have applied neural networks to similar asset pricing problems that traditional factor models have previously considered. They have been able to make more accurate predictions with their approach compared to traditional approaches.

1.1 Problem formulation and motivation of the study

Size, value and *momentum* are well-known factors and many studies have investigated them (Fama and French 2012; Carhart 1997; Asness et al. 2018; Asness et al. 2013). Today, there are also many investment vehicles that try to capture the benefits arising from these factors. Even though these factors are well researched in the literature, analysing them with neural networks may provide value to the empirical asset pricing literature. Neural networks may capture the nonlinear connections between these factors and returns that have been previously hidden. This then could lead to more accurate models for empirical asset pricing. These can be valuable for future research and market practitioners. Making accrued predictions is always important to researchers and can help them to better understand problems. More accrued forecasts can shed light on asset pricing mechanisms and allow more precise risk premium measurement. Understanding the risk premium is a fundamental problem in asset pricing research.

Market practitioners can also benefit from more accrued forecasts. They can be used for market timing, selecting securities, risk diversification and assessment of true out-performance. This is why this study forms neural network based factor models that predict returns. This study also forms a comparable linear model for return prediction. Once these models are formed, the study can evaluate the benefits of using neural networks as a basis of factor-related return predictions. The study also formulates investment strategies that could be traded and are based on return predictions obtained from the neural network based factor models. By investigating investment strategies that can be traded, economic performance, meaning and significance can be evaluated. This economic evaluation provides another consideration on the benefits of neural network based factor models. In this study model evaluation methods and performance analysis are chosen so that it provides value to both researchers and market practitioners.

Deep learning models might be under-utilised in the empirical asset pricing context by both practitioners and researchers. There are only a few studies that consider deep learning in a similar context to traditional factor models. The complexity of neural network models might be one reason why deep learning is not very popular in empirical asset pricing. Its black-box nature coupled with the need for large datasets might also be reasons why they are not used in the literature. For practitioners the same reasons apply as to why they might not use deep learning. Practitioners can also have rules that prevent them from using black-box methods and they might deem them riskier, as the exact logic with a certain prediction is unknown. Given the potential of deep learning models and their under-utilisation it is interesting to apply them to empirical asset pricing problems.

Previous studies that have considered asset pricing in the deep learning context have focused on a large number of characteristics. This study uses neural networks to price assets based on a few predetermined factors. An approach that considers a few predetermined factors may provide a different kind of understanding about factors and help practitioners and researchers understand asset pricing better. Limiting factors to only a few has been very common in the previous literature and it has raised some points that support focusing on a few factors. Thus, considering a limited number of factors in the deep learning context is reasonable and might provide an interesting addition to previous studies. Moreover, in practice, investors often consider only a few factors in their investment strategies. Such an approach from practitioners further increases interest in the limited number of factors in the deep learning context.

1.2 Scope and research questions of the study

This study forms a *size, value* and *momentum* factor based empirical asset pricing model by utilising neural networks. These factors are well established and they are not just short-term fading anomalies. These factors have economic reasons and intuition behind them. Given the significance of these factors, it is interesting to investigate them through deep learning. It is interesting to see how neural networks represent these established factors. It would also be interesting to know if more advanced and powerful models are able to provide better predictions on how these factors affect equity returns. From these points of view, the study might provide valuable information about asset pricing.

This study starts answering questions by forming a basic linear model based on the above-mentioned factors. The purpose of the basic linear model is to form a base comparison for the neural networks based approach. Next, different neural network models are hyper-parameter tuned and trained to predict returns based on the above-mentioned factors. Based on the neural network model and linear model, this study aims to create more understanding about empirical asset pricing, size value and momentum factors and whether neural networks are a good approach in empirical asset pricing.

Based on the viewpoint above, this study forms two questions which focus on neural networks in empirical asset pricing. The first question consider the suitability of neural networks for empirical asset pricing. The second question considers the neural network model's representation of the factors. Firstly this study forms a basic linear model and a neural network model to answer the following question:

- How well does the neural network model predict stock returns and how does it compare to the linear model?

Secondly, this study investigates the neural network's representation of the factors and answers the following question:

- How does the neural network interpret size, value and momentum factor model?

Multilayer perceptron (MLP) networks are used in this study to form the neural network based factor model. To understand the neural network's representation of asset pricing, the relation of different factors to return is investigated. This study also uses a temporal attention-augmented bilinear neural network model to gain more understanding on how a neural network sees an empirical asset pricing problem considering size, value and momentum. The data used in this study is United States stock market data and fundamental data obtained from Quandl.

This thesis proceeds as follows. The next chapter, ”Evaluation of empirical asset pricing”, considers the theoretical framework of the study and previous literature related to it. In addition, the chapter presents the models that this study uses to form the empirical asset pricing models. The third chapter focuses on the study's setup and the performed analysis. After that the final chapter presents the conclusions of the thesis.

2 Evaluation of empirical asset pricing

Empirical asset pricing is the practice of estimating expected returns. The target is to estimate expected returns so that they reflect the actual returns as well as possible. Another way to look at empirical asset pricing is to see it as measurement of the risk premium. According to the rational asset pricing view, all returns are caused by risk premiums. Therefore, estimating security returns through the lens of the risk premium leads to the goal of explaining all the returns. Factor models have a long history in financial literature concerning empirical asset pricing. With factor models a security's characteristics are used to explain and predict its returns. The characteristics or group of characteristics are seen as a factor, like a value, that affects the security's returns. From the risk premium perspective, factors are sources of risk premium. Fama and MacBeth (1973) first started to explain security returns with risk. Later Fama and French (1993) formed their three-factor model which takes into account risk, size and value factors. After the first factor models, the financial literature has seen many studies on factor models. The financial literature has considered various different factors, combinations of factors and markets when factor models are concerned. More recently, more sophisticated machine learning techniques have been introduced in factor models. For example Gu et al. (2018) explore multiple machine learning models as the basis for a factor model.

2.1 Empirical asset pricing

2.1.1 Factor model

Empirical asset pricing with factors is the practice of predicting future returns based on variation in some attribute. *CAPM*, presented in Equation 2.1 below, can be seen as a factor model with one factor. Given *CAPM* the expected security return is Sharpe (1964)

$$r = \alpha + \beta r_m, \tag{2.1}$$

where r is the expected return and β is the factor for market risk, r_m is the market return and α is the extra return that cannot be explained.

Fama and French (1993; 1996) added more attributes to *CAPM* and formed their three-factor model. Their model predicts returns based on three attributes,

$$r_i - r_f = \alpha + \beta_i R_m + s_i SMB + h_i HML. \tag{2.2}$$

In the equation beta, s and h are the factor loadings for market risk, size and value, respectively. SMB, HML and R_m are the expected risk premiums for size, value and market. Carhart (1997) proposes a four-factor model where momentum is added to the model similarly to size and value. In the same way Fama and French (2015,2016) form their five-factor model, which helps to explain return better than their three-factor model. In a similar fashion many more models and factors have been formulated in the literature. Factor models with more factors or different factors can be formed with similar logic.

Below, Equation 2.3 presents a general factor model with lagged characteristics. The factor model can be combined this way with any factors and any number of factors. Here the expected return for the next time period is expressed as follows:

$$R_{i,t+1} = E(v_{i,t}\lambda) + \alpha, \tag{2.3}$$

where $v_{i,t}$ is the vector of factors in given stock i at time point t. λ denotes the transformations from vectors to returns. (Gu et al. 2018)

The model is formed by solving optimisation where the α is minimised. This leads the model to predict returns as well as possible. If the model could predict returns perfectly would always be zero. Equation 2.4 below presents the target function for a general factor model when the mean squared error of the predicted return is optimised. In other words, the target function for the factor model is to minimise the squared summed α

$$\Upsilon(\lambda) = \frac{1}{N}\frac{1}{T}\sum(r_{i,t+1} - E(v_{i,t}, \lambda))^2 \tag{2.4}$$

This is the target function that is minimised when pricing an asset empirically. This target functions holds for linear and nonlinear models.(Gu et al. 2018)

Two main forms of empirical asset pricing are studying cross-sectional models and time-series models. The cross-section approach focuses on the attribute's effect on the security's return. With the cross-section approach it is usually assumed that the factor's risk premium or in other words its slope is constant, thus the security's expected return depends only on the attribute which measures the factor exposure under consideration. In the empirical asset pricing context, time-series factor models usually consider time-changing risk premiums or in other words factor loadings. This means that the expected return not only depends on its exposure to the factor under consideration but also on the time of the exposure. With Eq. 2.2 this would mean that with the cross-section approach SMB, HML and R_m would be constant over time and with the time-series approach they would change over time. (Fama and French 2019) In other contexts forming a model over time with many securities and lagged variables could be seen as combining both the cross-sectional and time-series approaches, but with asset pricing this is not usually the case. Eq. 2.3 and 2.4 hold if the model is considered cross-sectional or time-series or both.

2.1.2 Linear factor model

The factor model presented above is formed by finding the values for factors so that the alpha is minimised. The smaller the alpha the better the model can explain returns. The traditional linear way of finding values for factors is to run regressions. For example in Fama and French's 3-factor model R_m, SMB ja HML premiums and the sensitivities for these premiums can be found by running regression over given data on securities attributes and returns. Forming these kinds of regression models with lagged variables has been the main way to predict asset prices, thus empirically pricing assets. (Fama and French 1996; Fama and French 1993; Gu et al. 2018)

A common approach to investigate factors and build factor-based asset pricing models is to use sorted portfolios. In this approach decile portfolios are formed based on some characteristic. Based on the differences in the portfolios' returns the impact of the characteristic on returns can be estimated. Similarly to single-characteristic portfolios, portfolios can be formed based on two or more characteristics. These sorted portfolios can then be used to explain returns and build asset pricing models. Traditional asset pricing models are formed similarly to Equations 2.2 and 2.3 using these sorted portfolios and regression. Sorted portfolios and regression define the risk premiums and factor loadings. For example, this method is used to investigate returns based on size, value and momentum and to build asset pricing models based on them. These models have been able to capture pricing anomalies that CAPM does not capture. (Fama and French 1992; Fama and French 1996; Fama and French 2015)

Traditional factor models that have a large number of factors do not explain returns significantly better than models considering only a few factors. One reason for this is that traditional models handle large sets of factors poorly. Also only a handful of factors provide significant predictive gains. Studies have shown that more advanced models also depend on a few factors to provide the most information. Some factors are able to capture the risk premia of other factors (Fama and French 1996). Although the literature has recognised hundreds of factors, only a handful are useful when it comes to forming asset pricing models. (Gu et al. 2018; Feng et al. 2019; Green et al. 2017) For these reasons it is reasonable to investigate asset pricing models consisting of a few factors.

Creating a linear factor model by using regression or regression and a sorted portfolio is a simple and traditional way to make a factor model. However, these models have their weaknesses. Studies have combined different approaches with traditional regression. One way to improve the simplest model is to use time-varying risk premiums thus combining cross-sectional and time-series models (Fama and French 2019). It is also possible to enhance a linear model with more advanced statistical techniques and to try to avoid the pitfalls of a linear model (Gu et al. 2018). Traditional linear models have had a lot of success with empirical asset pricing but their explanatory power over return is by no means perfect. This leaves room for more advanced models when empirical asset pricing is considered.

2.1.3 Deep learning in asset pricing

Some previous studies have applied machine learning techniques to asset pricing to better understand asset returns. Machine learning uses advanced statistical models and focuses on predicting outcomes. These features of machine learning mean that it is attractive to apply to asset pricing. It may also help to overcome some of the difficulties found with more traditional approaches to asset pricing.

The earliest application of machine learning in finance concerns the forecasting of the price of derivatives. Hutchinson et al. (1994), Garcia et al. (2000) and Gencay and Qi (2001) have used neural networks to price and hedge derivative securities. Yao et al. (2000) and Wang (2009) among many others have used deep learning to forecast derivative prices. Studies by Rapach et al. (2013) and Leaung et al. (2000) have used machine learning techniques to solve forecasting problems in finance.

Machine learning has also been applied successfully in high-frequency trading context. Ntakaris et al. (2018) use neural networks among other methods to predict price movements in high-frequency limit order books and achieve promising results. Ntakaris et al. (2019) make predictions in similar context with multiple neural networks showed that selected features and relatively simple MLP networks can achieve promising prediction results. Nousi et al. (2019) again achieve good results with MLP networks and high-frequency limit order books when price movements are considered. Mäkinen et al. (2019) use convolutional long short-term memory network architecture with attention to predict high-frequency limit order book data. This network, that can consider time series features and focus on the most import instances, predicts jumps better than previous network architectures. Passalis et al. (2018) use another neural network architecture that can capture times series properties. Their temporal-aware neural bag-of-features model gives again promising results in limit order book context.

Neural networks have been successfully used to make stock return related predictions. They have been used to make similar predictions to the Fama and French (1993) type of asset pricing problem. They have also been used to make different kinds of predictions with stocks. These applications come from different literature but show the potential of neural networks and stock return prediction. Neural networks are able to make relevant time-series predictions, classifications and approximations on stocks. (Heaton et al. 2017; Heaton et al. 2016; Fischer and Krauss 2018)

Making factor models and studying cross-sectional asset pricing similar to Fama and French (1993) was not in the forefront of using machine learning in finance. Now studies are starting to apply machine learning techniques to do characteristic-based stock return forecasting. Gu et al. (2018) have used multiple machine learning techniques to predict stock returns. They found that these methods can provide a better forecast for returns. In their study the best performing methods are *trees* and *neural networks*. Gu et al. trace their better predictive power to the ability to capture nonlinear interactions.

Feng et al. (2018) created a deep learning stock characteristics-based factor and conditional factor model. Like Gu et al. (2018), Feng et al. use a multi-layer neural network to form their factor model. The Feng et al. model generates hidden deep factors with a neural network from public stock characteristics. In addition to this cross-sectional analysis they add state conditions to explore bull-bear factor exposures. Their neural network based model offers significant forecasting improvement over CAPM and the Fama-French three- and five-factor models.

There is potential to make similar return prediction models to those of Fama and French with neural networks. Naive approaches do not work but advanced methods may provide meaningful results. Neural networks have been applied to Fama-French problems in a way where a large set of characteristics are used to make predictions. The model trains itself to select the most important characteristics. (Gu et al. 2018; Feng et al. 2018) It remains unanswered how neural networks perform when they are given a smaller number of factors that are chosen in advance. This approach would be more similar to the traditional way of doing empirical asset pricing. This takes away one advantage that deep learning has over traditional models, in that it can handle a large amount of variables. It still leaves the advantage of allowing non-linearity.

Although neural networks offer significant advantages over traditional approaches, there are some challenges in using them in asset pricing problems. The Fama-French type of asset pricing problems have a low signal-to-noise ratio. These problems also have a relatively small amount of data compared to other problems where neural networks have excelled. Another challenge is a stable pattern of non-stationary asset return prediction. These issues make it harder to create accurate models for asset pricing. (Gu et al. 2018; Feng et al. 2018)

Deep learning can be used to solve theoretical and practical problems in finance. Deep learning provides more accurate predictions than traditional models. Deep learning can model complex non-linearities and reveal connections that linear models cannot capture. Smaller prediction error leads to better understanding of asset pricing. Identifying economic structures more precisely and the ability to price assets with small pricing errors may challenge current views of market efficiency and other theories.(Heaton et al. 2016; Gu et al. 2018; Feng et al. 2018) Neural networks are thus a useful way to model and research traditional empirical asset pricing problems to gain more accurate information.

2.1.4 Size, value and momentum

Size, value and momentum are factors commonly used in asset pricing literature. From the Fama-French (1992) model, size and value are used to explain asset return. The size factor is usually given by the market equity of a given stock (ME). The value factor considers the actual value and the price of owning that value. A stock's book equity divided by its market equity (BE\ME) is the usual measure by which value is determined. Size and value factors are able to capture some variation in cross-sectional stock returns. The portfolios of small firms tend to have higher returns than the portfolios of larger firms. Portfolios with high Book-to-Market values tend to have higher average returns than low BE\ME values. Value and size are significant factors and evidence of their return premiums is strong in various markets. (Fama and French 1992; Fama and French 1993; Fama and French 2015)

Economic intuition behind the size premium considers smaller firms to be riskier than larger ones. Smaller firms then have a higher cost of capital, thus they need to return more. The reason Why smaller firms are riskier has been theorised to be due to growth options. It is suggested that the market value measure classes the more riskier firms as smaller. More riskier firms have a lower market value if everything else is equal which leads to the market value measure to categorising riskier firms as smaller firms by definition. Size premium has also been explained in relation to liquidity risk. Studies have also theorised behavioural reasons for the size effect. (Asness et al. 2018)

The economic reasoning of the value premium considers value firms to be riskier. Value firms are riskier because they have more assets in place and the assets in place are hard to reduce. This makes assets in place riskier than a growth option, particularly in tough times. (Zhang 2005) It has been shown that value firms assets' betas and their leverage rise, leading their market risk exposure to rise. Growth firms are much less sensitive to this kind of rise in market risk exposure. (Choi 2013)

Momentum was quickly added to Fama and French's (1993) 3-factor model to explain returns. Adding this price change related factor meaningfully reduced pricing errors and thus better explained returns. Momentum reveals that stocks that have higher previous returns tend to have higher future returns.(Fama and French 2012; Carhart 1997) There is strong evidence of a momentum premium, where past winners earn more in the future, in different markets. (Asness et al. 2013; Rouwenhorst 1998)

There is no clear risk-based reason why momentum should explain returns. Some studies have tried to connect momentum to other factors and explain phenomena that way. Yet its connection to returns is well recognised in the literature. The lack of a clear risk reason and strong evidence has led to behavioural and information processing intuitions about momentum. Models based on these intuitions have been able to explain momentum patterns that are otherwise hard to justify. Momentum patterns are explained by periods of underreaction and overreaction. Some studies caution against overly relying on this explanation. Models that lead to underreaction anomalies and do not require irrational behaviour, heterogeneous information or market friction have also been suggested in the literature. (Hong and Stein 1999; Lee and Swaminathan 2000; Jegadeesh and Titman 2001; Johnson 2002; Rouwenhorst 1998) Even though there may not yet be a full economic explanation for momentum it is a strong phenomenon, probably linked to behavioural elements.

Factors do not exist in isolation of each other. There are correlations between them. Controlling one variable and exploring another shows how, in some cases, the behaviour of one factor varies in relation to another. Size has been shown to alter the slope and magnitude of other factors. With some factors like momentum, factors can have opposite effects on returns when it comes to different size deciles. There are recorded interactions between size and value, momentum and quality. (Asness et al. 2014; Fama and French 1996; Fama and French 2012; Fama and French 2015; Fama and French 1992) Similar connections and correlations to size and momentum and value also exist between other factors. Strong connections between momentum and value and their patterns have been found in multiple markets (Asness et al. 2013). If these connections among factors are nonlinear and asymmetric, the neural network approach can provide relevant value over traditional linear models. Neural networks can capture these connections, whereas traditional linear models do not capture these connections to their full effect.

As the literature recognises hundreds of factors, some studies have tried to identify which of them truly matter. Green et al. (2017) investigate 94 characteristics and find 12 that provide independent information about returns. One of these characteristics is value related and two are momentum related. Feng et al. (2019) developed a procedure to assess which factors provide extra value given some existing factors. They noticed that only few actually provide extra value. These kinds of findings support the approach where only a few factors are considered in the model.

Neural network representations of stock returns and characteristics find momentum, liquidity, risk and value ratio factors to be the most influential predictors of returns. One-month momentum seems to be the most important characteristic. Six other momentum characteristics are among the top twenty most influential characteristics, i.e. momentum change, recent max. return, 12-month momentum, 6-month momentum, industry momentum and 36-month momentum. The second most important characteristic is log market equity. Other liquidity characteristics are among the top 20 most influential characteristics. Risk characteristics related to volatility and beta are also influential. Earning increases and sales-to-price are fundamental and valuation ratio characteristics belonged to the top 20. (Gu et al. 2018)

Size, value and momentum factors are well established in the literature and there is strong evidence that these factors have a connection to returns. Early asset pricing models found these factors and they also seem to be meaningful in newer, more sophisticated models. These factors also seem to have characteristics that the traditional approach to factor models might not capture, thus exploring machine learning models with these factors might lead to better predictions.

2.2 Models

2.2.1 Modelling task for empirical asset pricing

The models in this thesis solve regression and classification tasks. The models are linear models or neural network models. The linear model solving regression task is a standard ordinary least squares regression. The neural networks that consider the regression task are MLP neural networks. Models with a regression task are formed with a similar target of minimising unexplained returns. Models minimise α by minimising the means square error between the prediction and realised returns following Eq. 2.4.

The linear model is formed here by following Eq. 2.4. The model follows the standard ordinary least squares (OLS) method. OLS performs regression over securities over time to find the optimal λ for Eq. 2.4. The optimal λ minimises the sample pricing error between average predicted return and actual return.

The models that solve the classification task consider the problem similarly to decile portfolios. This study's classification does not form decile portfolios simply based on ranking attributes but rather uses attributes to predict whether a given stock is going to be a future winner, neutral or a loser. Models solving classification tasks minimise cross entropy. Eq. 2.5 presents the loss when the classification task is solved:

$$L(x,p) = -\sum_{k=1}^{K} y_k ln(p_k), \tag{2.5}$$

where k is the class, K is the number of classes, y is the binary class label, p is the predicted probability and x is the input (Graves 2012 pp. 15-17). In this study the temporal attention augmented bilinear model is a neural network model concerning the classification task and thus considers Eq. 2.5 as a loss. The linear model solving the classification task is logistic regression. Logistic regression solves the following optimisation:

$$min_{w,c} w^T w \frac{1}{2} + C \sum_{i=1}^{n} ln(exp(-y_i(x_i^T w + c)) + 1) \tag{2.6}$$

where w represents the weight, y the output, x the input, c is the regularisation parameter and n the number of observations (Scikit-learn 2021).

2.2.2 Neural network

Neural networks are an example of supervised learning methods. In supervised learning the correct output value for the training example is known. Based on these known values, the models optimise themselves to produce a certain output given a certain input. Neural networks are a deep learning method. Neural networks can learn very complex functions because they have multiple representation levels. Different representations are made up of simple but non-linear modules that transform the representation into a higher, more abstract level. How different levels represent a problem are not designed by humans but learned from the data used. Learning utilises only general-purpose learning procedures to change its internal parameters and develop the representations. (LeCun et al. 2015) Figure 2.1 presents a simple feedforward neural network. The network has four inputs, on hidden layer with three units and one output. Neural networks can have cyclical connections within them which are referred to as feedback, recursive or recurrent neural networks. (Graves 2012 pp. 15-35)

2.2.3 Multilayer perceptron

Probably the most commonly used feedforward neural network is the multilayer perceptron (MLP). An MLP consists of the perceptron processing units presented in Figure 2.2. The perceptron takes in the weighted inputs and bias. Then they are summed and feed the sum to the activation function. (Graves 2012 pp. 15-35) The activation function is denoted in Figure 2.2 as φ. It takes in the weighted sum of the variables and bias. A popular non-linear function is the rectified linear unit (ReLU). ReLU is defined as

$$ReLU = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

where z is the output and summed inputs. Other activation functions, like the sigmoid and softmax are possible. (LeCun et al. 2015) The sigmoid function is

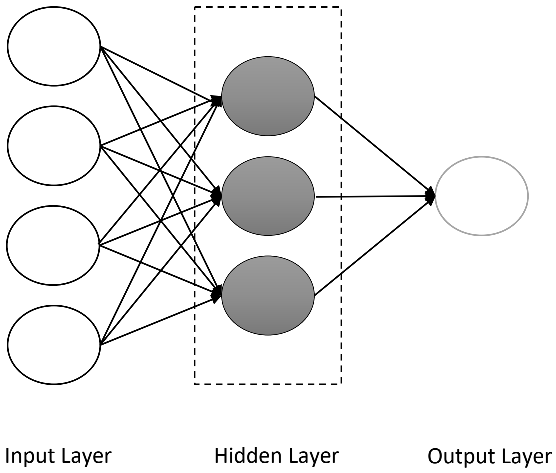

Figure 2.1 *Simple feedforward neural network with four inputs, one hidden layer with three neurons, and one output.*

described by the following function

$$sigmoid = \frac{1}{(1 + exp(-x))} \tag{2.8}$$

where x is the input for the function (Graves 2012 pp. 14). Tanh is another possibility and Eq. 2.9 below gives the definition of tanh as

$$tanh = \frac{(exp(x) - exp(-x))}{(exp(x) + exp(-x))} \tag{2.9}$$

with x being the input (Graves 2012 pp. 14).

A neural network learns by adjusting its internal parameters, usually called weights. These weights determine what kind of output is produced by a given input. The learning algorithm adjusts weights by computing the gradient vector for each weight. The gradient vector indicates how much the error would change if the weight increased by a tiny amount. The weights are then adjusted in the opposite direction to the gradient vector. To train a network, in another words to adjust the weights, two procedures are used: the forward and backward pass. Figure 2.3 presents these procedures. In feedforward the weighted sum of the previous layer is used as input. The previous layer's outputs are processed as inputs by the next layer's neurons. Neurons process the inputs as shown in Figure 2.2. In a backward pass the signal travels from outputs to inputs. Computed error derivatives (or gradients) can be passed to lower levels by the practical application of chain derivative

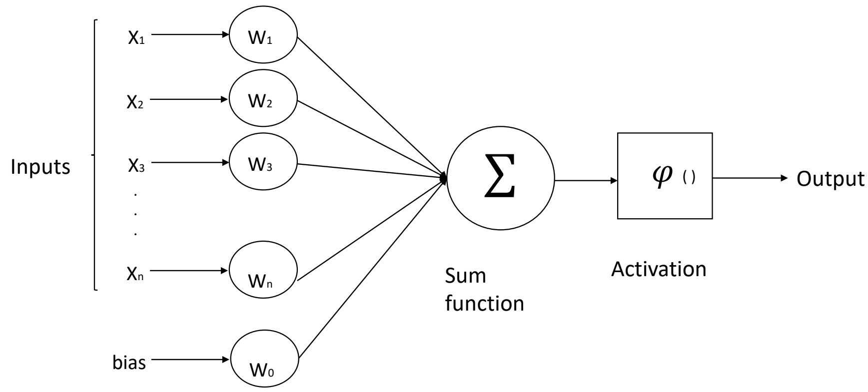

Figure 2.2 *Perceptron. (Ruck et al. 1990)*

rules. (LeCun et al. 2015)

Adam is a lower-order moment estimate based algorithm for the first-order gradient-based optimisation of stochastic objective functions. It is an efficient stochastic optimisation that requires only first-order gradients and little memory. *Adam* uses estimates of the first and second moments of the gradients to compute individual learning adaptive learning rates for distinct parameters. *Adam* combines AdaGrad (Duchi et al. 2011) and RMSProp (Tieleman and Hinton 2012) to work well with both sparse gradients and on-line and non-stationary settings. Program 2.1 below presents the pseudo-code for *Adam*. The algorithm requires the following arguments: α stepsize, $\beta_1, \beta_2 \in [0, 1)$ exponential decay rates for the moment estimates, $f(\theta)$ as the stochastic objective function with parameters θ and θ_0 as the initial parameter vector. Good default values for the parameters are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. (Kingma and Ba 2015)

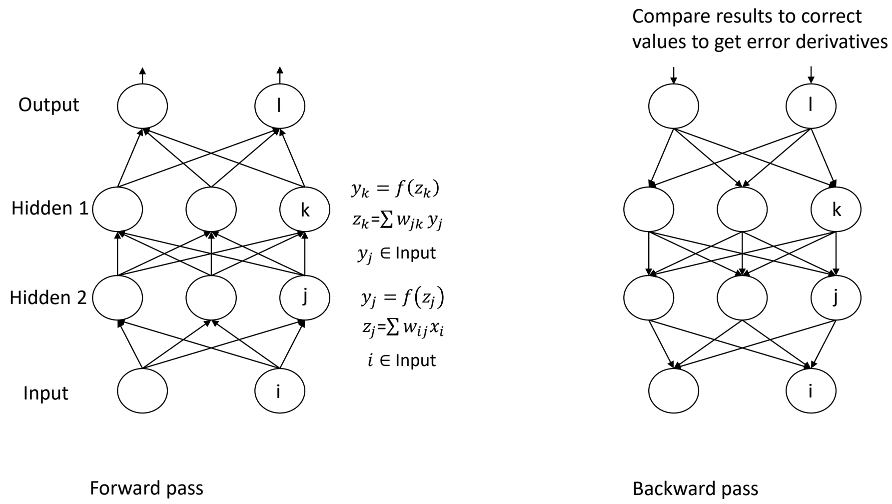

Figure 2.3 *Neural network forward and backward pass. In forward pass the signal travels towards the output. Total input z is the sum of the weights and the previous layer's outputs. Z is then given to the layer's activation function to produce outputs. In backward pass the signal travels in the opposite direction and error derivatives (or gradients) are passed between layers. (LeCun et al. 2015)*

Program 2.1 *Adam, stochastic optimization algorithm (Kingma and Ba 2015)*

Adam(α, β_1, β_2, $f(\theta)$, θ_0)
m_0 := 0 (Initialize 1^{st} moment vector)
v_0 := 0 (Initialize 2^{nd} moment vector)
t := 0 (Initialize timestep)
while θ_t not converged **do**
  $t \leftarrow t+1$
  $g_t \leftarrow \nabla_0 f_t(\theta_{t-1})$ (Get gradients respect to objective at time t)
  $m_t \leftarrow \beta_1 * m_{t-1} + (1-\beta_1) * g_t$ (Update biased first moment estimate)
  $v_t \leftarrow \beta_2 * v_{t-1} + (1-\beta_2) * g_t^2$ (Update biased second raw moment estimate)
  $\widehat{m_t} \leftarrow m_t/(1-\beta_1^t)$ (Compute bias-corrected first moment estimate)
  $\widehat{v_t} \leftarrow v_t/(1-\beta_2^t)$ (Compute bias-corrected second raw moment estimate)
  $\theta_t \leftarrow \theta_{t-1} - \alpha * \widehat{m_t}/(\sqrt{\widehat{v_t}} + \epsilon)$ (Update parameter vector)
end while
return θ_t (Resulting parameters)

2.2.4 Temporal attention-augmented bilinear network

Attention has been introduced in a variety of deep learning tasks. In many cases the introduction of attention has been beneficial. Previous work has applied attention in the context of financial time series. This previous work has produced promising results with financial time series. Tran et al. (2019) proposed a temporal attention bilinear network for financial time-series data predictions. The proposed network is applied here to the factor model problem.

A bilinear layer (BL) transforms inputs to output by preforming the mapping presented in Equation 2.10 below:

$$Y = \phi(W_1 X W_2 + B) \tag{2.10}$$

where $\phi()$ is an element-wise nonlinear transformation function like the ReLu presented in equation 2.7 . W_1, W_2 and B are the parameters that are to be estimated. X is the bilinear layer's input. Given that $X \in \mathbf{R}^{D \times T}$ is a second-order tensor with D and T being dimensions of the first and second mode, then $W_1 \in \mathbf{R}^{D' \times D}$, $W_2 \in \mathbf{R}^{T \times T'}$ and $B \in \mathbf{R}^{D' \times T'}$. The mapping transforms the input into a matrix of size $D' \times T'$. The bilinear layer has an advantageous property when time-series modelling is concerned, which is that the BL layer's transformation learns dependencies for each mode of the input representation. (Tran et al. 2019)

Introducing temporal attention to bilinear network allows the network to learn the importance of each time instance. The BL layer presented above cannot learn this from the representation. A temporal attention-augmented bilinear (TABL) layer maps the input $X \in \mathbf{R}^{D \times T}$ to output the following way:

$$\bar{X} = W_1 X \tag{2.11}$$

$$E = \bar{X} W \tag{2.12}$$

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})} \tag{2.13}$$

$$\tilde{X} = \lambda(\bar{X} \otimes A) + (1 - \lambda)\bar{X} \tag{2.14}$$

$$Y = \phi(\tilde{X} W_2 + B) \tag{2.15}$$

where $\otimes$ is an element-wise multiplication operator. a_{ij} and e_{ij} are elements in positions of i and j in A and E. ϕ denotes the nonlinear transformation function again. Y is the output and belongs to $\mathbf{R}^{D' \times T'}$. The TABL estimation parameters are $W_1 \in \mathbf{R}^{D' \times D}$, $w_2 \in \mathbf{R}^{T \times T'}$, $W \in \mathbf{R}^{T \times T}$, $B \in \mathbf{R}^{D' \times T'}$ and λ. λ is between one and zero, both included. The temporal attention-augmented bilinear layer inherits two dependencies from the bilinear model with a different semantic meaning. It does this with W_1 and W_2. In addition to this, TABL learns an intermediary attention step with W and λ parameters. The forward pass which produces Y for TABL is presented in Figure 2.4. (Tran et al. 2019)

In the forward pass presented in Figure 2.4 above, steps two, three and four apply the attention mechanism to the layer. The attention mechanism stimulates

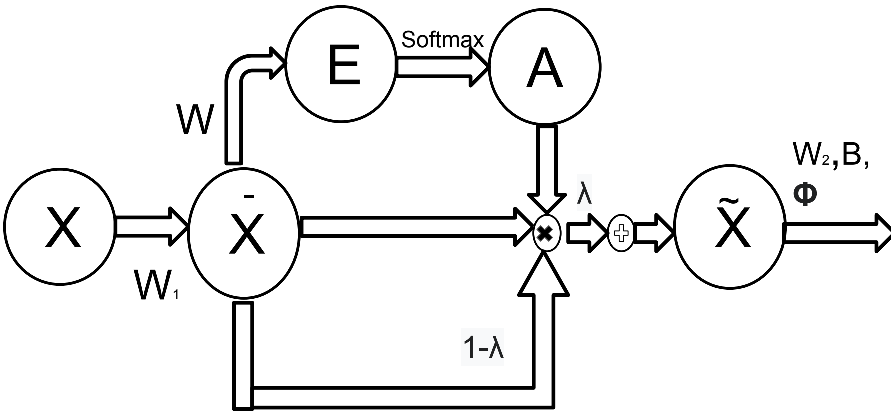

Figure 2.4 *Temporal attention-augmented bilinear layer forward pass that produces output. The forward pass consists of the following steps: 1) Application of Eq. 2.11, thus turning each column of X into a new feature space. 2) The second step applies Eq. 2.12 to the process with the aim of bringing the relative importance of the temporal instances towards each other. 3) In the next step the attention mask A is produced by applying Eq. 2.13 to the last step's result E. 4) Next the attention mask is used to nullify unimportant elements. This happens by applying Eq. 2.14 where learnable parameter λ makes the soft attention mechanism possible. Using soft attention instead of hard attention helps the model to learn in the early stages. 5) The final step applies Eq. 2.15 to the process and produces the higher level representation. (Tran et al. 2019)*

competition between neurons representing different temporal steps of the same feature. A TABL layer can be trained by applying back-propagation. (Tran et al. 2019) The key advantage of the TABL model is its ability to focus on certain parts of the time series through attention. With this attention mechanism it also possible to see what the model gives prominence to. Interpreting what the model thinks is important sheds lights on the black box predictions and might give insights into the phenomena behind the time series.

2.2.5 Dropout technique

Overfitting is a common problem with neural networks. A deep neural network with multiple hidden layers can learn very complex relationships between their inputs and outputs. Because of limited training data some of those complex relationships will be the result of sampling noise. Those connections are only in the training set, not in the real test data. One way to deal with overfitting is to combine many different neural nets at test time. Often, combining several models is not a feasible solution because it requires a lot of computation and data. Dropout is a technique that addresses the issue of overfitting and problems of combining models. With dropout units, both hidden and visible units, are temporarily removed from the network together with all their incoming and outgoing connections. The choice of which units are dropped from the network is random. In the simplest instance, each unit is retained with fixed probability. The retraining probability can be chosen based on the validation set or be simply set to 0.5. This value seems to be close to the optimal value for various neural networks and tasks. (Srivastava et al. 2014)

Using the dropout technique leads a neural network to sample a thinned network from itself. The thinned network consists of all the units that pass through the dropout. The new thinned network is sampled and trained for each presentation of each training case. Training a neural network with n units and dropout can be seen as training a 2^n collection of thinned nets with extensive weight sharing. Each thinned network is trained very rarely, if ever. Explicitly averaging the prediction from exponentially many thinned models is not viable, but a very simple approximate averaging method works well in practice. At test time a single neural network without dropout is used. The weights of this network are scaled-down versions of the trained weights. At test time the outgoing weights of a unit are multiplied by the probability that the unit will be retained in training. This leads to the situation where any hidden units' expected output is the same as the actual output at test time. By applying this scaling, networks with shared weights can be combined into a single net. Training a neural network with dropout and using this approximate averaging at test time leads to significantly lower generalisation error. (Srivastava et al. 2014)

3 Empirical analysis of return predictions

This chapter presents the empirical analysis of the study and continues as follows: first data used in analysis is presented. Then the study's setup and evaluation metrics are presented. After that the regression task results and final neural network models concerning this task are presented. Next, the classification task and attention analysis are considered. Together with these results the final neural network models concerning this task are also presented. After that the economic meaning of predictions obtained from neural networks is considered. Finally the limitations for the analysis are presented.

3.1 Data

The data used for this thesis is NYSE and Nasdaq listed stocks. The time period for the data is between January 2009 and October 2019. The data consists of daily observations, totalling 10 million observations. The number of different stocks in the data is over 7100. The average number of stocks per day exceeds 4200. The average number of observations for a particular stock is more than 1400. Data observation has to have a value for all the factors and return measures to be included in the data set. Stock data was obtained from Quandl (www.quandl.com). The data used in this study is from Quandl's Sharadar Core US Equities Bundle dataset. Price data was obtained from Sharadar Equity prices data table and the price-to-book ratios and market capitalisations were obtained from the Sharadar Core US Fundamentals data table. The data was then combined based on the data's ticker and date. For prices the study uses the adjusted stock price from Sharadar Equity prices. Summary statistics for the data are presented in table 1 in Appendix B.

The three factors used in the study are formulated in the following way from the above-mentioned data. The size factor is measured by the firm's market capitalisation. Value is the firm's price-to-book ratio. Price-to-book is the stock price divided by the book value per outstanding share. For momentum this study uses multiple indicators. All of the momentum indicators are based on the adjusted stock price, logarithmic returns and trading days. This study uses the following momentum indicators: one week, one month, three months, six months and twelve months, which are calculated as the past 5, 21, 63, 126 and 252 trading day returns. Trading days here are days that exist in the data for stock. One additional momentum indicator is used which combines the one month and 12 month momentum indicators. This momentum is the standard momentum used in the literature which is the last twelve months' returns minus the last month's returns. The risk-free interest rate, obtained from the four-week T-Bill rate reported by the U.S Department of the Treasury available on their website (home.treasury.gov) that corresponds to the momentum indicator, was subtracted from the value of the momentum indicator.

Returns are logarithmic cumulative returns over the following 21 trading days. The returns that are calculated are excess returns over the risk-free interest rate. For the classification section, returns are divided into three classes. Stock that belongs to the winner class has returns in the top 30 percentile of all returns. The loser class consists of stocks that have returns in the bottom 30 percentile. The third class consists of stocks that have returns between the top and bottom 30 percentiles. As returns are stock-specific returns in 21 trading days, the bottom and top classes are quite evenly distributed over the years in the data.

3.2 Methods

3.2.1 Experimental setup

This study uses the following test setup. First of all the non-categorical variables were scaled between -1 and 1 for all models. Scaling usually helps learning and is standard procedure when deep learning problems are concerned. Secondly, the study uses anchored walk forward for training and evaluation.

Empirical analysis in this study used the following procedure for training, validating and testing all models. Initial hyper-parameter tuning and training was done with data consisting of observations before the year 2015. The procedure uses a train-validation split with 80 percent for training and 20 percent for validation. This split is a common rule of thumb in machine learning. After the initial hyper-parameter tuning phase the procedure moves to training and testing phase. During the training, in this phase, 5% of training data is held for validation. Data held for validation is used to determine when to stop training and thus prevent overfitting. Training is stopped when prediction performance in validation data stops improving.

In this study, empirical analysis used anchored walk forward evaluation for testing model performance. Models were trained with data that contained observations from year y_0 to year y_t and then tested with the following year's y_{t+1} observations. The training data y_0 to y_t contains a validation part that is used to determine when to stop training. Figure 3.1 presents this anchored walk forward training and testing procedure. The first testing year was 2015. This means that the analysis has five different folds to run and investigate models. Given the scarcity of data this approach seemed the appropriate one to apply.

For the linear models parameter tuning was ignored and a simple OLS model was chosen for regression predictions and simple logistic regression was chosen for

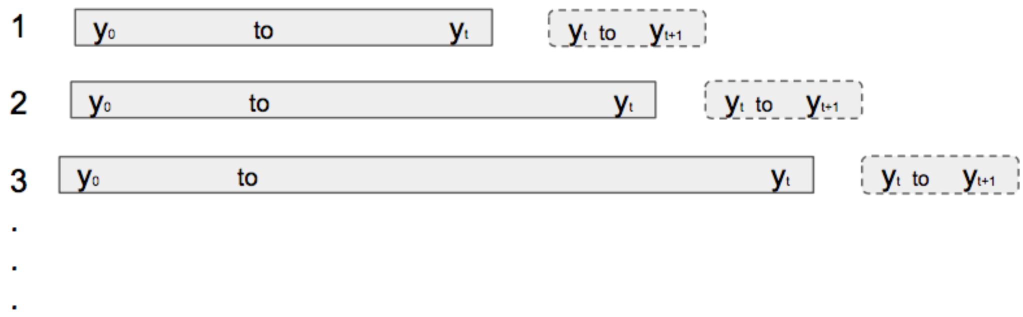

Figure 3.1 *Anchored walk forward process where the dashed line box is the test set and the solid box is the training set containing the validation part. Ascending numbers represent the increasing value for t.*

the classification task. Using logistic regression as the linear model for the classification task and thus the benchmark model for the attention model seemed the most appropriate, given the regression task linear model and the nature of the classification task studied here. Logistic regression is a simple linear model for solving classification tasks and thus seemed to be an appropriate benchmark model to replace the OLS model in the classification task. This study also considered using the sorted portfolio approach in some capacity as the classification task's linear model but logistic regression seemed more fitting for this study.

For the neural network models various hyper-parameters were considered. The noisy nature of stock return data came up during the hyper-parameter tuning phase. It proved hard and time consuming to find the right parameters so that the neural network models would be feasible to continue to the next phase of the study. Different network sizes and shapes were considered. In addition, different optimisers and activations were considered. Hyper-parameter tuning was iterative. First, larger differences were tried and then the most promising was selected for another round of iteration. Iteration continued until this study arrived at its final models. For MLP and Attention models the three most promising were selected for evaluation. In addition to this, one attention model with different input shape, thus different attention, was selected. These final networks are presented later together with their performance results. The selection was based on mean squared error or accuracy in validation. For regression tasks the mean squared error was used and for classification tasks the accuracy was used to measure success.

3.2.2 Performance evaluation

To assess the predicting performance of the models used, the study employed different measurements used in previous studies concerning out-of-sample return predictions. Measurements are calculated from logarithmic returns. The study used a few different measurements to form an accurate assessment of the models' performances and differences in performance. The first evaluation measurement is the root mean squared error, which is computed as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{r}_i - r_i)^2}, \tag{3.1}$$

where $\hat{r}$ is the predicted return and r is the actual return. Feng et al. (2018) use this measurement to assess the prediction performance of their neural network based factor models. R^2 statistic is another another method to assess out-of-sample performance. It allows comparison between the performance of two forecasts. Feng et al. (2018), Campbell and Thompson (2007) and Gu et al. (2018) use R^2 to evaluate their out-of-sample performances. Out-of-sample R^2 is calculated as

$$R^2_{Comparison} = 1 - \frac{\sum_i^N (r_i - \hat{r}_i)^2}{\sum_i^N (r_i - r_i^{'})^2}, \tag{3.2}$$

where r_i is the actual return, $\hat{r}$ is the predicted return and $r^{'}$ is another predicted return. $R^2_{Comparison}$ is used in this study when a model-to-model comparisons are considered.

Campbell and Thompson (2007) use historical means as a comparison value. While comparison to historical means is popular, it is flawed according to Gu et al. (2018) when individual stock returns are concerned. They argue that using historical means lowers the bar for good forecasts because historical mean stock returns are noisy. Using zero as a comparison forecast value avoids the drawback of using historical returns. This study evaluates forecast performance with zero forecast. Out-of-sample R^2 with zero return is calculated as

$$R^2_{Zero} = 1 - \frac{\sum_i^N (r_i - \hat{r}_i)^2}{\sum_i^N (r_i)^2}, \tag{3.3}$$

where r_i is the actual return, $\hat{r}$ is the model predicted return. Equation 3.3 holds when logarithmic returns are considered. R^2_{Zero} is used to evaluate the model forecasts.

When classification problems are examined in this study the F-Score is used as the metric to evaluate the models' performance. Classification problems considered in this study are problems where there are multiple classes. The micro average version of F-Score:

$$F_{micro-average} = 2\frac{precision_{micro} * recall_{micro}}{precision_{micro} + recall_{micro}}, \tag{3.4}$$

where $precision_{micro}$ is the sum of true positives dived by sum of true positives and false positives. $Recall_{micro}$ is the sum of true positives dived by sum of true positives and false negatives. $F_{micro-average}$ seems to be the most fitting metric as it considers the overall prediction performance and gives more weight to more common classes. These features make it fitting given the study's classification problem.

3.3 Regression prediction analysis

3.3.1 Linear-regression

This study uses two OLS-based models as the baseline comparison for the study's neural network models. OLS models apply naively standard OLS regression to form the study's linear models. This approach is in line with the traditional approach to forming factor models. Two variations of the OLS-based factor model consider different input variables. One, called OLS-All here, uses all the input variables considered in this study. The OLS-All model makes its excess return predictions based on market value, price-to-book ratio and 5, 21, 63, 126 and 252 trading day returns. The other linear model, called OLS-standard here, takes a more standard approach to input variables. The size and value inputs here are standard inputs used in the literature when these factors are considered. Using multiple periods for momentum is not the usual approach in the literature as far as momentum is concerned. The standard momentum in the literature is the past year's return minus the last month's return. In this study the OLS-standard model uses this definition of momentum. The OLS-standard, which is very in line with the standard academic approach, uses market value, price-to-book ratio and the 252 trading day return minus the last 21 trading day returns as its inputs.

The OLS-All model has an RMSE of 17.57% in the testing data. OLS-standard achieves 17.60% in the same measure. The R2 statistic for OLS-All is -0.11% and for OLS-standard the R^2_{Zero} statistic measure is -0.46%. These results indicate that the OLS-All model is slightly better at predicting excess returns for stocks. The R^2_{Zero} statistic shows that both of these models do a worse job of predicting excess returns than the naive prediction of zero return. Table 3.1 presents the previously mentioned RMSE results and the yearly RMSE results for all the models. Table 3.2 contains the corresponding R^2_{Zero} statistics. The tables reveal that there is variation in the RMSE and R^2_{Zero} statistic over the testing years. Both OLS-All and OLS-

Standard are able to beat naive zero prediction in some testing years, even though when all the testing years are considered they do not beat the naive assumption of zero return. Both OLS models have two years where the R^2_{Zero} statistic is below zero and three years where it is above zero. Even though there are more testing years where R^2_{Zero} is positive than when it is negative, the overall R^2_{Zero} for the OLS models is still negative. The negative years are so significant that the overall R^2_{Zero} for both OLS models is negative. The R^2_{Zero} results for OLS seem especially bad in one of the negative years compared to the other testing years. The years where the R^2_{Zero} is negative are the same for OLS-All and for OLS-Standard.

Table 3.1 *RMSE for OLS-based models as percentage.*

Year	OLS_All	OLS_Standard
2015	17.89%	17.86%
2016	17.85%	17.87%
2017	17.02%	17.13%
2018	17.33%	17.36%
2019	17.81%	17.82%
All	17.57%	17.60%

Table 3.2 *R^2_{Zero} for OLS-based models as percentage.*

Year	OLS_All	OLS_Standard
2015	1.78%	2.04%
2016	-1.79%	-1.99%
2017	-3.67%	-5.03%
2018	1.44%	1.10%
2019	1.74%	1.53%
All	-0.11%	-0.46%

3.3.2 Multilayer perceptron

After the initial validation three MLP models were selected for testing. These models are similar in other ways but the number of neurons in their hidden layer changes. The models are called MLP 64, MLP 128 and MLP 256 as they have the corresponding number of neurons in their hidden layers. They all have two hidden layers containing the model-specific number of neurons. The models also use batch normalisation and dropout. The dropout rate for the models is 0.2. The optimisation is Adam, as mentioned above. Activation is ReLu for the layers except for the output layer activation, which is sigmoid. These MLP models are presented in Figures 1, 2 and 3 in Appendix A.

Table 3.3 *RMSE for multilayer perceptron models as percentage.*

Year	MLP_64	MLP_128	MLP_256
2015	17.94%	17.89%	17.82%
2016	17.79%	17.88%	17.71%
2017	16.67%	16.68%	16.73%
2018	17.29%	17.24%	17.36%
2019	17.74%	17.76%	17.76%
All	17.48%	17.49%	17.47%

Table 3.4 R^2_{Zero} *statistics for multilayer perceptron models as percentage.*

Year	MLP_64	MLP_128	MLP_256
2015	1.18%	1.72%	2.50%
2016	-1.15%	-2.12%	-0.22%
2017	0.51%	0.35%	-0.21%
2018	1.86%	2.40%	1.09%
2019	2.45%	2.19%	2.25%
All	0.89%	0.84%	1.06%

Table 3.3 presents the RMSE results for the MLP models and table 3.4 presents the R^2_{Zero} statistics. MLP 64 has an RMSE of 17.48%. For MLP 128 the RMSE is 17.49% and for MLP 256 the RMSE is 17.47%. The R^2_{Zero} values for MLP 64, MLP 128 and MLP 256 are 0.89%, 0.84% and 1.06%, respectively. Given these R^2_{Zero} results the MLP models beat the naive zero return prediction and thus can be said to explain the stocks' excess returns. The MLP model containing the most parameters seems to be the best at predicting excess returns, having the smallest RMSE and the largest R^2_{Zero} statistic. Looking at the yearly results in the tables some variation between the testing years can be found. MLP 256 has two years when its R^2_{Zero} is below zero, indicating poorer prediction ability than naive zero return prediction. MLP 64 and MLP 128 have one year when R^2_{Zero} is less than zero. Whereas MLP 256 has two negative testing years where R^2_{Zero} is about -0.2%, MLP 64 and MLP 128 have one testing year where R^2_{Zero} is much more negative compared to MLP 256 below zero R^2_{Zero} testing years.

3.3.3 Model comparison

Regarding the R^2_{Zero} statistic and RSME results presented above for the MLP, and OLS models, it is easy to view MLP models as the most promising group. Moreover, Table 2 (Appendix B) presents a model-to-model comparison of $R^2_{Comparison}$ statistic results for the models. From these results we can compare the model's capability to explain excess returns over another model. A negative value indicates that the model has a poorer ability to explain variations than another model. A positive value indicates a better ability to explain variation in returns compared to another model. In Table 2 we can see that all the MLP models do better than the other models. MLP 64 and MLP 256 have values over 1% when compared to OLS models and MLP 128 has 0.95% value against OLS-All and 1.30% value when compared to OLS-Standard. Based on the results shown in Table 2, it is logical to select MLP models as the best models to predict stocks' excess returns. The results in Table 2 also show that the MLP 256 model explains the returns better than the two other MLP models and is found in this study to be the best model to explain excess returns.

Table 3.5 contains all the RMSE for all the models. Analysing the RMSE further strengthens the view that the MLP models are the best group of models for predicting excess returns in this study's setup. All of the MLP models tend to outperform the OLS models almost every year.There are a few testing years when some of the MLP models have a higher RMSE than one or both of the OLS models but in general the MLP models tend to outperform the OLS models consistently. MLP 64 has one year where it has higher RMSE than both of the OLS models. MLP 128 has also one year where its RMSE is higher than both of the OLS models. MLP 128 has also one year where its RMSE is higher to one OLS model and equal to another OLS model. MLP 256 has one year where its RMSE is equal to one OLS model and higher than the other OLS model. MLP 64 has higher RMSE, when worst performance year is compared, than both of the OLS models. For MLP 128 this worst RMSE is higher than one OLS model and equal to the other one. Worst year for MLP 256 is better than the worst years for the OLS models. Comparing the best prediction years for the models, it can be seen that all the MLP models have lower RMSE than those of the OLS models. The RMSE results show that the MLP models outperform the OLS models quite constantly. This is particularly true for MLP 256. Only thing weakening the argument for the MLP models is that the two MLP models have worse worst years than one or both of the OLS models. Nevertheless, the MLP models overall more accurate predictions and the frequency of more accurate prediction years make them more desirable group of models.

Table 3.5 *RMSE for all models as percentage.*

Model	2015	2016	2017	2018	2019	**All**
OLS-All	17.89%	17.85%	17.02%	17.33%	17.81%	**17.57%**
OLS-Standard	17.86%	17.87%	17.13%	17.36%	17.82%	**17.60%**
MLP-64	17.94%	17.79%	16.67%	17.29%	17.74%	**17.48%**
MLP-128	17.89%	17.88%	16.68%	17.24%	17.76%	**17.49%**
MLP-256	17.82%	17.71%	16.73%	17.36%	17.76%	**17.47%**

MLP-based factor models were the best models for predicting excess returns in this study's setup. MLP 256 was the best model to predict excess returns. The other two MLP models also outperformed the OLS models. This result is in line with the previous literature. Previous studies have shown that neural network-based factor models can outperform traditional linear regression-based factor models where prediction accuracy is concerned. Because MLP models were the best models in this setup, the author arrived at a similar conclusion about neural network-based factor models.

3.4 Classification and attention analysis

3.4.1 Classification results

With temporal attention models (TABL) this study predicts whether a given stock belongs to the winner, loser or neutral class. The study's benchmark model for TABL models is logistic regression, here referred to as log-reg. Log-reg predicts similar classification tasks as the TABL models in this study. The benchmark model was changed to log-reg because the prediction task was a classification task and log-reg is more suitable for that task than previous OLS models would have been. The TABL model allows this study to look inside the black box of the neural network and investigate whether its perspective on the problem is considered in this study. Looking at how a TABL model sees factors is the most important part of this study's analysis where TABL models are concerned. In this analysis the accuracy of the benchmark model and TABL model are still compared because the analysis results are much more interesting if the TABL model can be more accurate than its benchmark model.

A log-reg model has similar inputs to the OLS-Standard model. Its inputs are size, value and momentum where the last month is subtracted. Table 3.6 shows the F-Scores for models considered with the classification problem. From the table we can see that log-reg has an F-Score of 42.94%. Figure 6 in Appendix B presents the confusion matrix for log-reg model predictions. The figure indicates that log-reg does not really predict winners. Loser prediction is also much more rare than neutral prediction.

TABL models have a input of size, value and one month momentum as twelve month time-series. All TABL models use a dropout of 0.2. TABL 128 has a hidden layer of 128x6, TABL 256 has a hidden layer of 256x6, TABL 516 has a hidden layer of 516x6 and TABL Flipped has hidden layer of 128x3. The TABL models are similar to each other except when it comes to hidden layer units, expect TABL Flipped where the input is shape is turned so that attention can focus on input representing factors and not temporal instance. The same activations and optimisation are used in this study as in Tran et al. (2019) . Figures 4 and 5 in Appendix A present the structure for different TABL networks. TABL 128 has an F-Score of 44.55%. For models TABL 256 and TABL 516 the F-Scores are 43.98% and 43.81% respectively. TABL Flipped has F-Score of 42.07%. Figures 7, 8, 9 and 10 in Appendix B present confusion matrices for TABL model predictions. From these figures we can see that predicting winners is hard and mixing winners and losers is common.

Table 3.6 $F_{micro-average}$*-Score % for models.*

Year	log_reg	TABL_128	TABL_256	TABL_516	$\text{TABL}_F lipped$
2015	42.84%	46.27%	45.42%	45.55%	43.12%
2016	42.30%	39.88%	38.99%	39.55%	39.17%
2017	45.52%	46.32%	45.87%	45.17%	43.22%
2018	40.88%	44.82%	44.72%	44.11%	43.08%
2019	43.26%	45.87%	45.37%	45.05%	41.75%
All	42.94%	44.55%	43.98%	43,81%	42.07%

Comparing the different models' F-Scores in Table 3.6 it can be seen that TABL 128 has the highest F-Score over all the testing years and in every testing year except testing year 2016. TABL 256 and TABL 516 also have a higher overall F-Score than log-reg. Both models have a lower F-Score than log-reg testing year 2016, similarly to TABL 128. TABL 516 also has a lower F-Score than log-reg in testing year 2017. However, TABL Flipped has lower overall F-Score than log-reg. TABL Flipped has years when it has higher F-Score than log-reg and it has years when the score is lower. Given the results shown in Table 3.6, the TABL models, except TABL Flipped, seem more accurate at predicting whether any given stock is a winner, loser or neutral than the log-reg model. TABL 128 seems the most accurate at this classification task.

Furthermore, examining the confusion matrices for the models in Figures 6, 7, 8 and 9 (Appendix B), a few observations can be made. Firstly, the TABL models seem to have fairly similar results to each other and quite different compared to log-reg results. TABL models also tend to favour the neutral prediction but not as extremely as log-reg. More predictions in the neutral category is logical as 40% of returns belong to this class, whereas the loser and winner class each have 30% of returns belonging to them. Log-reg seems to have a hard time predicting anything other than the neutral category. TABL models do not have the same problem. Log-reg made a few predictions for the loser class and almost none for the winner class. TABL models also favour the loser class over the winner class but make a much more reasonable amount of predictions for both classes. TABL models seem to differentiate neutral stocks quite well from winners and losers. They have more trouble in deciding, if a stock is not neutral, whether it is a winner or loser. As losers are favoured by these models over winners, many winners are classified as losers.

3.4.2 Attention analysis

To gain insight into the black-box of neural networks this study calculated the average attention values for the TABL 128 model and TABL Flipped. Attention values are captured during model training from an attention mask, A in Fig. 2.4. The attention values analysed here are captured from the last epoch for the TABL 128 model that is trained with data before 2019 so its test period would be 2019. When attentions are captured the model is not trained exactly the same as it was trained for making the predictions. The batch size is different between the training occasions with the final epoch and also epoch number differing. This is due to the method how neural networks attention values were captured. The model's performance and thus predictions should be similar. The TABL 128 model was chosen for the attention analysis because it was the most accurate at classifying stocks into winners, losers and neutrals in this study's test. TABL Flipped was chosen because it allows attention to focus on different factor instances, where TABL 128's attention focuses on different time instances. The test period and training phase were chosen so that the network would have the chance to learn as much as possible.

Figure 3.2 presents average attention values for winners, neutrals and losers from TABL 128. A few clear patterns emerge with the presented attentions. With neutral category there is a clearly much higher focus on one instance. T-2 has much higher focus than other instances when determine neutral category. T-6 has some average attention of 0.13 which quite low compared to t-2, which has average attention of 0.85. Other attentions for neutral category are close to zero. Winner and loser attention patterns differ from neutral's attention pattern, but winner and loser attention patterns are similar between themselves. Attentions are focused on every other instance quite evenly. The not focused once being close to zero. T-2 instance somewhat lower than the other once that are among the focused instances.

Different attention patterns in 3.2 between neutral category and other categories and quite similar attention patterns between winner and loser categories makes sense when the classification results presented above are considered. The classification results showed that models were better at predicting neutral stocks but had more trouble predicting which class, winner or loser, the stock belonged to. Therefore it makes sense that the TABL 128 model was not able to find different things to focus on between winners and losers.

Attention pattern similarities and differences between categories make economic sense and indicate that the TABL 128 model was able to find focusing instances that make a stock riskier. Even though the TABL 128 model was able to find what to focus on when differentiating between safer and riskier stocks, it had more

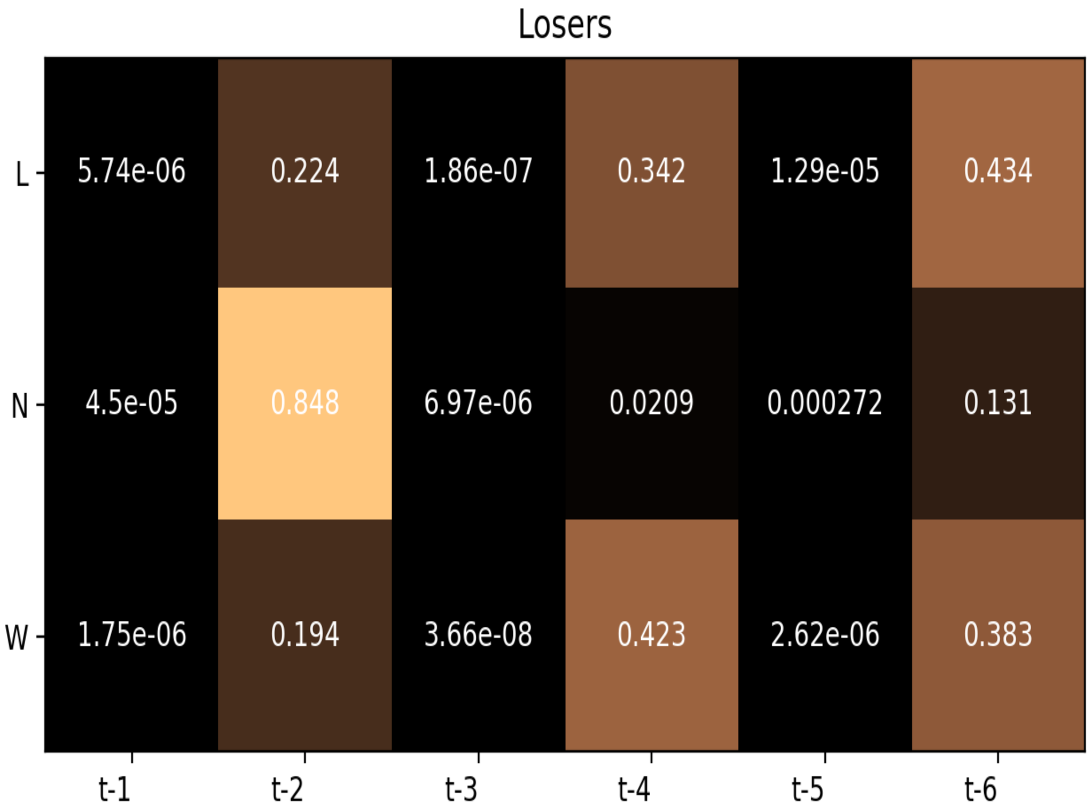

Figure 3.2 *Attention values for different temporal instances. Columns present the different temporal instances. Rows represent the different categories where L is the losers, N is the neutrals and W is the winners.*

difficulty recognising whether the risk would lead to higher gain or loss. This fits with classical economic thought about higher risk having higher rewards. Riskiness entails difficulty in determining whether stock is going to go up or down, otherwise it would not be truly risky. Same information might be important for both winners and losers and they also might be hard to differentiate as the actual classification result indicate. Given this, it is logical that the TABL 128 model focuses instances similarly with winner and losers.

Attentions to different factor instances obtained from TABL Flipped are presented in Figure 3.3. With all categories attentions are evenly distributed over all the factor instances. These results would indicate that all the factor instance thus the factors are important for the model. It is also notable that the model from which these attentions are captured was not particularly good at predicting categories. Given this, it is possible that the model was not able to learn focusing and all the factors are not as evenly important. Nevertheless, these factor instance attentions seem to indicate that all factors considered here are important for the model.

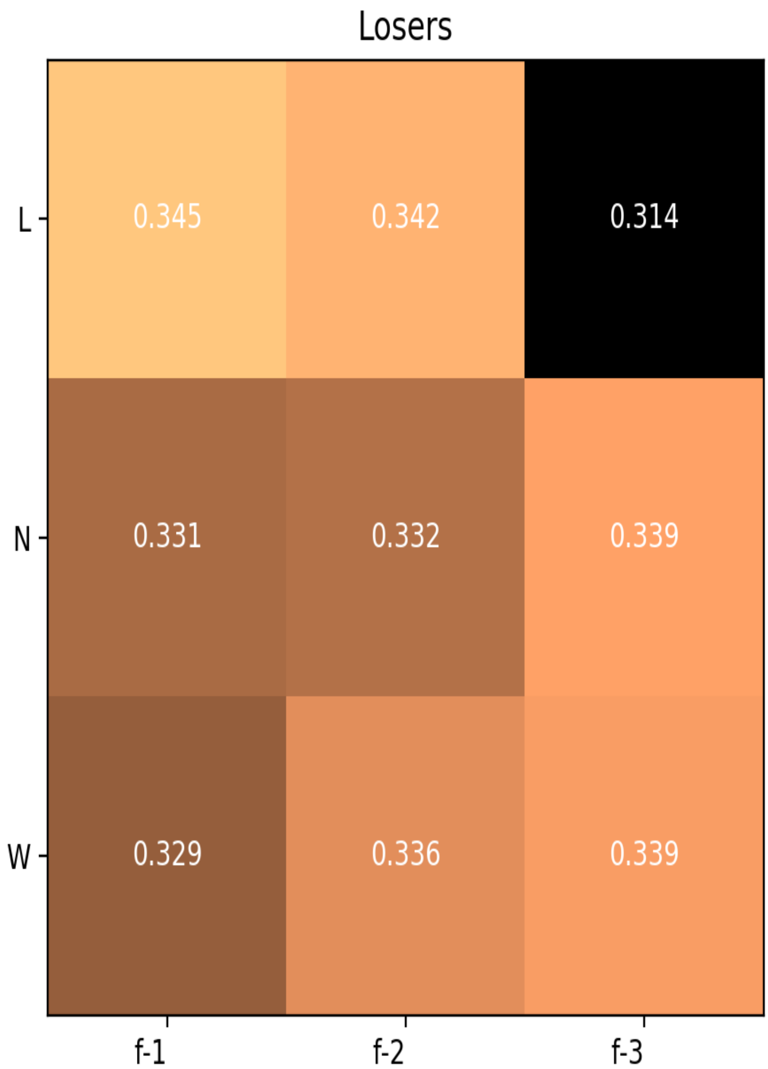

Figure 3.3 *Attention values for different factor instances. Columns present the different factor instances. Rows represent the different categories where L is the losers, N is the neutrals and W is the winners.*

3.5 Economic analysis

So far this analysis has shown that neural networks can be used to form factor models. An MLP-based factor model is more accurate at predicting returns than naive linear regression based approaches. The MLP-based factor models also beat zero prediction in their accuracy. This study's TABL-based factor model was also able to beat its linear benchmark. Knowing that these models are able to produce a prediction about stock returns raises the question of what is the economic significance of these neural network-based factor models. This study formed portfolio-based neural network models and tested them in a testing dataset used to determine the prediction accuracy of the model.

The economic analysis examines the most accurate models from the previous analysis. The analysis forms strategies based on the MLP 256 and TABL 128 models and tests their performance in the study's testing dataset. Figures 3.4 and 3.5 present box-whisker plots for monthly excess return for portfolios formed either from MLP 256's predictions or from TABL 128's predictions and also for a portfolio containing all of the stocks. The portfolios are equally weighted and sum up to 1. The monthly return for the portfolio is then the average of its stocks' monthly excess returns.

Figures 3.4 and 3.5 show that both MLP 256 and TABL 128 seem to have the ability to predict returns in a way that has economic meaning. Stocks that the MLP-factor model predicts will have a positive return form a portfolio that tends to do better than a portfolio that contains all stocks. This Positive-MLP portfolio has a median monthly excess return of 0.93% compared to a median of 0.64% for a portfolio of all stocks. Averages for the Positive and All portfolios are 0.65% and 0.46%, respectively. A portfolio formed of the stocks that MLP 256 predicts will have a negative return tends to perform more poorly than a portfolio of all stocks. The monthly excess return median and average for a portfolio of predicted negative returns are 0.30% and 0.15%. respectively. These results show that selecting stocks based on an MLP-factor model can enhance returns. Table 3 in Appendix B contains further statistics for MLP portfolios.

The TABL factor model also makes a prediction that can help the investor to select stocks. A portfolio of predicted neutrals has a higher median and average monthly excess return than a portfolio consisting of all stocks. A portfolio of predicted losers does worse than a portfolio of all stocks. A portfolio of losers has a lower median and average monthly return. A portfolio that has neutral predictions also has smaller dispersion in its monthly returns than a portfolio of all stocks. The neutral portfolio's returns seem more stable compared to the other portfolios. The neutral portfolio has a higher median and average than a portfolio of all stocks. The winner portfolio's returns vary a lot. It has the lowest median of the portfolios but the second highest average. It is notable that the winner portfolio consists of very few stocks at times. At times the TABL model does not predict many winners and in general it favours predicting other classes. This makes the winner portfolio

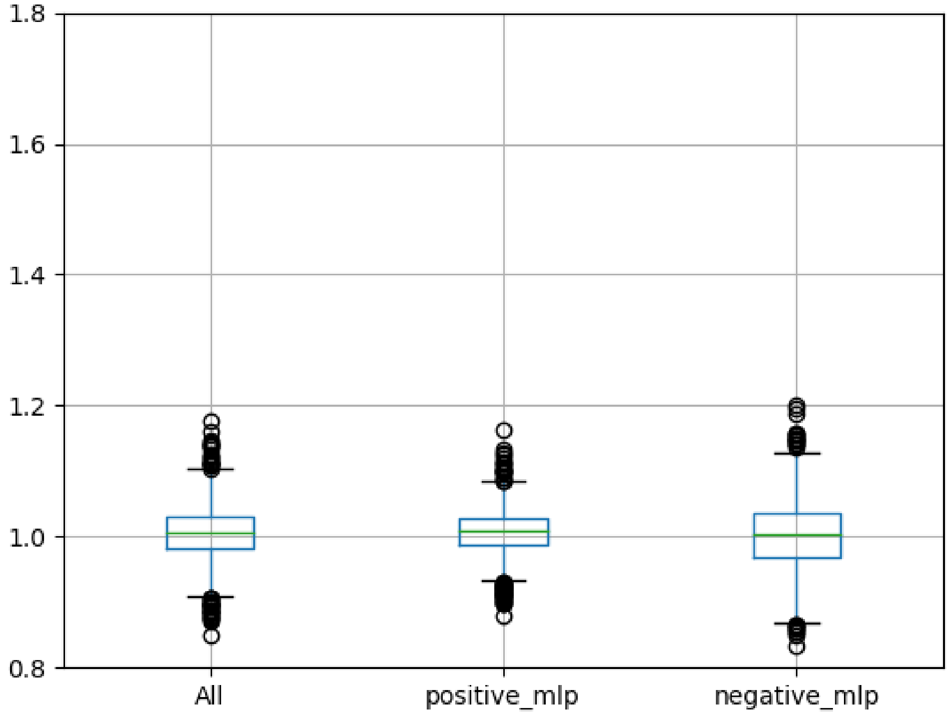

Figure 3.4 *Box-and-Whisker plot for MLP portfolios' monthly excess returns. The Y-axis is the real monthly excess return. The All-portfolio is an equally weighted portfolio of all stocks on a given day. The Positive-MLP portfolio contains those stocks for which MLP 256 predicts positive or zero returns on a given day. The Negative-MLP portfolio contains those whose MLP 256 predictions are negative. Both MLP portfolios are equally weighted. The All-portfolio median monthly return is 0.64% with an average of 0.46%. For the Positive-MLP the median and average are 0.93% and 0.65%. The Negative-MLP portfolio has a median monthly excess return of 0.30% with an average of 0.17%.*

slightly unreliable and the return results reflect that. It seems that with the TABL model real economic gains can be made by excluding losers and picking stocks that are predicted as neutral. Table 4 in Appendix B contains further statistics for TABL portfolios.

Figures 3.6 and 3.7 show the cumulative returns for MLP 256- and TABL 128-based strategies. This return is the return an investor would get by following these strategies, if trading and other implementation costs are ignored. Strategies start from the first available date in the testing period. Other possible starting dates are not considered. Using only one starting date avoids overfitting and giving results that are overly optimistic. Practice has shown that optimising the starting date has a large effect on a strategy's returns. Strategies are re-balanced monthly, counted from the trading days.

From Figure 3.6 we can see that the investor would have earned a higher return

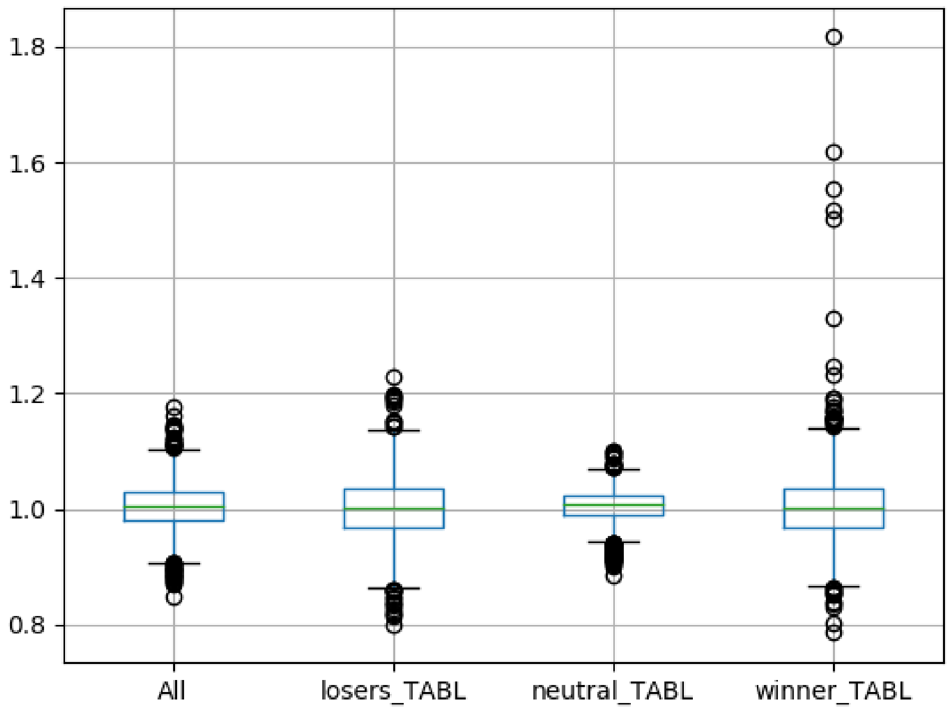

Figure 3.5 *Box-and-whisker plot for TABL portfolios' monthly excess returns. The Y-axis is the real monthly excess return. The All-portfolio is an equally weighted portfolio of all stocks on a given day. Loser, neutral and winner portfolios are formed based on the TABL model prediction. Stocks are equally weighted in these portfolios. The All-portfolio has a median and average of 0.61% and 0.47%, respectively. The Losers portfolio has an average monthly excess return of 0.10% and a median of 0.31%. The neutral portfolio has an average of 0.53% and median of 0.79%. For the Winner-portfolio the average and median are 0.51% and 0.22%, respectively. The test data for regression and classification slightly differ due to some missing data points, which is why the All-portfolio differs when compared to the TABL- and MLP-based portfolios.*

following the Positive-MLP strategy or MLP-LS strategy. The Positive-Strategy seems to have consistently a slightly better performance than a portfolio of all stocks and would seem to be a portfolio that an investor could benefit from. The MLP-LS strategy is not consistent like the Positive-MLP strategy, but in the end has a higher return. The Negative-MLP strategy has a lower return, as one would expect.

From Figure 3.7 we can see that the investor could benefit from following TABL-based strategies. In the end the neutral TABL strategy is the only strategy doing better than the portfolio of all stocks. The neutral portfolio seems to be constant in its performance. These results for TABL strategies are in line with previously considered return results. The same notion about investors' benefits from TABL predictions can be made here too. Investors can benefit by negatively selecting

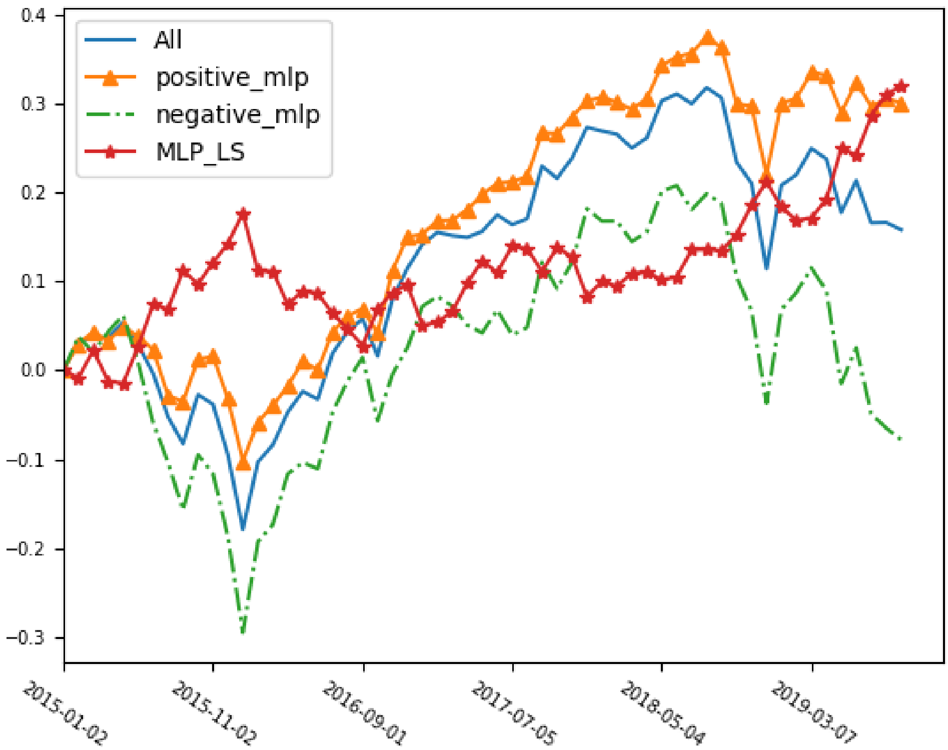

Figure 3.6 *Cumulative returns for MLP strategies. The chart presents logarithmic cumulative return strategies that are based on MLP predictions. Strategies re-balance monthly and start at the first available date in the testing period. The All-portfolio is a portfolio of all stocks with equal weighting. Positive-MLP is a portfolio of equal weighted stocks that MLP predicts will bring a positive monthly return. Negative-MLP is an equally weighted portfolio of the stocks that have predicted negative return according to the MLP model. MLP-LS is a long short portfolio of Positive-MLP and Negative-MLP. Negative-MLP is shorted with the portfolio value and a long position with the portfolio value is taken for Positive-MLP.*

losers from TABL predictions.

The return results here show that investors can achieve better returns by utilising neural network-based factor models. Practitioners can use neural network.based models for stock selection and market timing. Positive MLP and Neutral TABL seem to be attractive strategies for practitioners due to their consistency and higher returns. These results also warrant more research into neural network approaches and utilisation of their predictions in economic research.

3.6 Limitations

One limitations for this study is the size of the dataset. The dataset used is the largest that was available for this study, given resource constraints. Nevertheless,

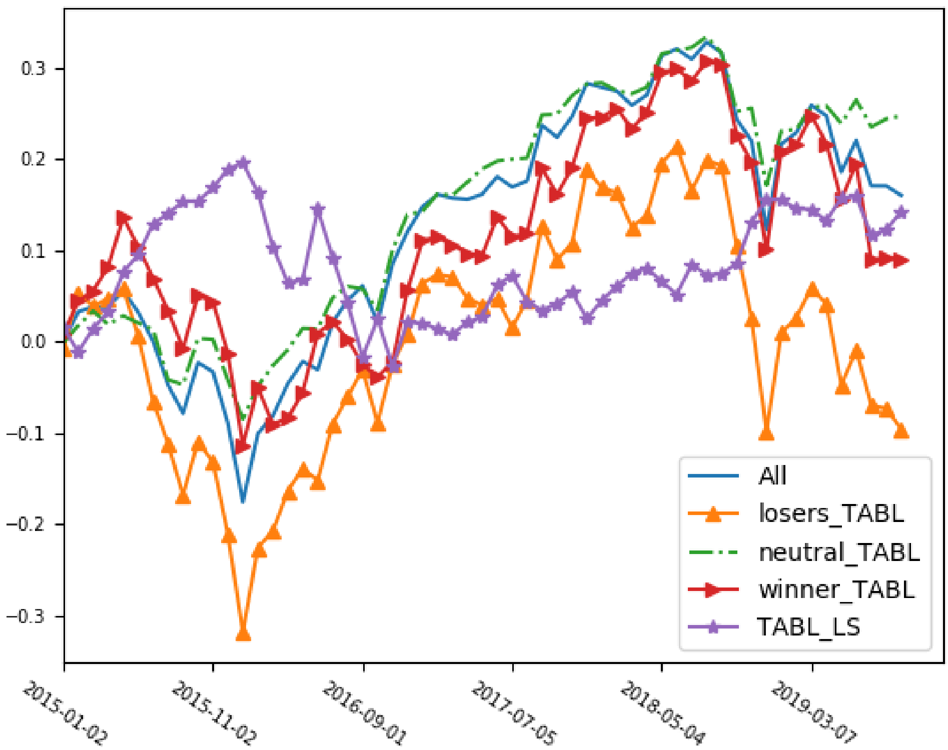

Figure 3.7 *Cumulative returns for TABL strategies. The chart presents logarithmic cumulative return strategies that are based on TABL predictions. Strategies re-balance monthly and start at the first available date in the testing period. Again the All-portfolio is the portfolio of all stocks with equal weighting. Losers are a portfolio of stocks that TABL predicted to be losers with equal weighting. Neutral is a similar portfolio of neutral predictions as the winner portfolio for winners. TABL-LS is a long short portfolio of predicted winners and losers. Losers are shorted with the portfolio value and a long position with the portfolio value is taken for winners.*

the dataset is relatively small. Particularly, the dataset has quite short a timeframe. Often studies that consider factors in empirical asset pricing use datasets that have multiple decades of data. Also studies that have used neural networks in similar empirical asset pricing context, have had datasets containing data from multiple decades. It is the opinion of the author here, that if testing data would contain at least ten years, it would increase the reliability of the results. This would allow testing to contain different economic conditions and market cycle phases, thus providing stronger evidence for using neural networks as basis of investment decisions. Because the study had a scarce amount of data it used observations that contain overlapping data in a sense that the future returns used contained days that overlapped with other future returns. The quality and thus reliability of the results could be improved by a larger dataset. If a much larger dataset were available, removing overlapping from the observations could also make the results more robust. Also using a more widely used dataset would enhance the reliability of this study's results. Using common dataset like COMPUSTAT that is well established and extensively used in asset pricing literature would increase the reliability of the results.

Another limitation of this study is the amount of computational power available. The computational capacity available limited the hyper parameter tuning and the number of different models that were tried. Quite a small number of different combinations were tried for different networks. Trying more combinations could have led to more accurate neural network models being found. This is especially true for TABL (TABL Flipped) model that have attention focused on factor indicators and not to the temporal instances.

Another consideration that could result in more convincing results is usage of better benchmark models. In this study the simplest approach to linear models was taken. Using models that are not so naive would make it harder for neural network models to beat their alternative. Constructing linear model, that performs better at considered prediction task, as benchmark would thus asses the benefit of using neural networks more accurately. Approaches from previous literature, like for example sorted portfolios or time varying risk-premium, could have been tried in effort to form better performing benchmark model. In addition to linear models, some previously used machine learning approach could have been used as benchmark model in effort of trying to construct better benchmark model. Lack of better benchmark models is probably the main limitation to this study and would strengthen the argument, that neural networks can enhance empirical asset pricing, the most.

4 Conclusions

Generally, factors are important consideration in today's equity markets and they are well recognised phenomena in the literature. Size, value and momentum are widely acknowledged factors. There is a strong empirical evidence and theoretical base for them in the literature. The traditional way to use factors in empirical asset pricing is to form a linear factor model with regression. A recent approach in a few studies has been to use neural networks as a base for factor models.

Although a few previous studies had considered neural networks in empirical asset pricing, the author of this thesis felt that they might be underutilised in Fama-French type of empirical asset pricing problems. This study set out to investigate the benefits of using neural networks in empirical asset pricing. Firstly, MLP neural networks were used in the regression task of predicting stock returns. This thesis answered the question of how well neural network models predict returns by calculating the RMSE for the models and comparing their prediction to a forecast of zero with the R^2_{Zero} statistic. The study's MLP 64, MLP 128 and MLP 256 had an RMSE of 17.48%, 17.49% and 17.47%, respectively, over the test sample. The MLP models outperformed the zero forecast prediction. This thesis also compared neural network models to traditional linear models. The MLP models were able to predict returns more accurately than linear models. These results considering MLP neural networks prediction performance and comparing those to linear models' corresponding ones, answered study's first research question "How well does the neural network model predict stock returns and how does it compare to the linear model?".

Second research question "How does the neural network interpret size, value and momentum factor model?", was answered by performing a classification task of predicting stocks with TABL neural networks and interpreting TABL neural networks' attention patterns. The F-scores for TABL 128, TABL 256, TABL 516 and TABL Flipped were 44.55%, 43.98%, 43.81% and 42.07% over the testing period. These F-scores were higher, except for the TABL Flipped, than comparable linear model F-scores, indicating better prediction performance. To investigate a neural network's representation of a size, value and momentum factor model average attention values for the best performing TABL model and TABL Flipped were computed in this thesis. The attention values revealed the capability to focus on different aspects to differentiate riskier stocks from less riskier. It also revealed that neural network focus similarly to temporal instances when top performers and bottom performers are considered. Attention analysis also seem to indicate that all factors considered in this thesis are relevant when stock returns are determined.

In addition, this thesis showed that these neural network predictions have economic meaning. Investment strategies that could be traded were formed based on neural network predictions. These strategies showed that neural network predictions can be used to obtain better returns than blindly investing in all stocks.

This thesis also makes a few key contributions. The results contribute to existing empirical asset pricing literature by showing the feasibility of the neural network approach when a limited number of factors are considered. The results confirm some previous observations about the neural network approach and imply that more frequent use of neural networks is warranted, at least if a naive linear more is considered as the other option. The thesis also showed that factor models can be analysed by utilising attention. Previously this approach had not been applied to Fama-French type problems but to a high-frequency setting. For practitioners this thesis shows that neural networks can be utilised in stock selection. The results also showed that the attention mechanism could be used to overcome some hurdles, such as the black-box nature of neural networks, which prevent practitioners from using them.

The study also revealed that there is room for future research. Future research could focus on applying neural networks that incorporate attention to wider and more extensively used datasets and use better benchmark models. Future research could also focus more on attention patterns and attention patterns where more factors are considered. Future research could also consider attention in regression predictions.

References

Asness, C. et al. (2018). "Size matters, if you control your junk". In: *Journal of Financial Economics* 129.3, pp. 479–509.

Asness, C. S., A. Frazzini, and L. H. Pedersen (2014). "Quality minus junk". In: *Review of Accounting Studies*, pp. 1–79.

Asness, C. S., T. J. Moskowitz, and L. H. Pedersen (2013). "Value and momentum everywhere". In: *The Journal of Finance* 68.3, pp. 929–985.

Campbell, J. Y. and S. B. Thompson (2007). "Predicting excess stock returns out of sample: Can anything beat the historical average?" In: *The Review of Financial Studies* 21.4, pp. 1509–1531.

Carhart, M. M. (1997). "On persistence in mutual fund performance". In: *The Journal of finance* 52.1, pp. 57–82.

Choi, J. (2013). "What drives the value premium?: The role of asset risk and leverage". In: *The Review of Financial Studies* 26.11, pp. 2845–2875.

Duchi, J., E. Hazan, and Y. Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization". In: *Journal of Machine Learning Research* 12.Jul, pp. 2121–2159.

Fama, E. F. and K. R. French (1992). "The cross-section of expected stock returns". In: *the Journal of Finance* 47.2, pp. 427–465.

— (1993). "Common risk factors in the returns on stocks and bonds". In: *Journal of financial economics* 33.1, pp. 3–56.

— (1996). "Multifactor explanations of asset pricing anomalies". In: *The journal of finance* 51.1, pp. 55–84.

— (2012). "Size, value, and momentum in international stock returns". In: *Journal of financial economics* 105.3, pp. 457–472.

— (2015). "A five-factor asset pricing model". In: *Journal of financial economics* 116.1, pp. 1–22.

— (Aug. 2019). "Comparing Cross-Section and Time-Series Factor Models". In: *The Review of Financial Studies* 33.5, pp. 1891–1926. ISSN: 0893-9454. DOI: `10.1093/rfs/hhz089`. eprint: `https://academic.oup.com/rfs/article-pdf/33/5/1891/33209809/hhz089.pdf`. URL: `https://doi.org/10.1093/rfs/hhz089`.

Fama, E. F. and J. D. MacBeth (1973). "Risk, return, and equilibrium: Empirical tests". In: *Journal of political economy* 81.3, pp. 607–636.

Feng, G., S. Giglio, and D. Xiu (2019). "Taming the Factor Zoo: A Test of New Factors". In: *Fama-Miller Working Paper*, pp. 17–04.

Feng, G., N. Polson, and J. Xu (2018). "Deep Learning Factor Alpha". In: *Available at SSRN 3243683*.

Fischer, T. and C. Krauss (2018). “Deep learning with long short-term memory networks for financial market predictions”. In: *European Journal of Operational Research* 270.2, pp. 654–669.

Garcia, R. and R. Gençay (2000). “Pricing and hedging derivative securities with neural networks and a homogeneity hint”. In: *Journal of Econometrics* 94.1-2, pp. 93–115.

Gençay, R. and M. Qi (2001). “Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging”. In: *IEEE Transactions on Neural Networks* 12.4, pp. 726–734.

Graves, A. (2012). “Neural Networks”. In: *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15–35. ISBN: 978-3-642-24797-2. DOI: 10.1007/978-3-642-24797-2_3. URL: https://doi.org/10.1007/978-3-642-24797-2_3.

Green, J., J. R. Hand, and X. F. Zhang (2017). “The characteristics that provide independent information about average us monthly stock returns”. In: *The Review of Financial Studies* 30.12, pp. 4389–4436.

Gu, S., B. T. Kelly, and D. Xiu (2018). “Empirical Asset Pricing Via Machine Learning”. In: *Chicago Booth Research Paper No. 18-04, 31st Australasian Finance and Banking Conference 2018, Yale ICF Working Paper No. 2018-09, Available at SSRN: https://ssrn.com/abstract=3159577*.

Harvey, C. R., Y. Liu, and H. Zhu (2016). “… and the cross-section of expected returns”. In: *The Review of Financial Studies* 29.1, pp. 5–68.

Heaton, J., N. Polson, and J. H. Witte (2017). “Deep learning for finance: deep portfolios”. In: *Applied Stochastic Models in Business and Industry* 33.1, pp. 3–12.

Heaton, J., N. G. Polson, and J. H. Witte (2016). “Deep learning in finance”. In: *arXiv preprint arXiv:1602.06561*.

Hong, H. and J. C. Stein (1999). “A Unified Theory of Underreaction, Momentum Trading, and Overreaction in Asset Markets”. In: *The Journal of Finance* 54.6, pp. 2143–2184. DOI: https://doi.org/10.1111/0022-1082.00184. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/0022-1082.00184. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/0022-1082.00184.

Hutchinson, J. M., A. W. Lo, and T. Poggio (1994). “A nonparametric approach to pricing and hedging derivative securities via learning networks”. In: *The Journal of Finance* 49.3, pp. 851–889.

Jegadeesh, N. and S. Titman (2001). “Profitability of momentum strategies: An evaluation of alternative explanations”. In: *The Journal of finance* 56.2, pp. 699–720.

Johnson, T. C. (2002). "Rational momentum effects". In: *The Journal of Finance* 57.2, pp. 585–608.

Kingma, D. P. and J. Ba (2015). "Adam: A Method for Stochastic Optimization". In: *ICLR*. URL: http://arxiv.org/abs/1412.6980.

LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep learning". In: *nature* 521.7553, p. 436.

Lee, C. M. and B. Swaminathan (2000). "Price Momentum and Trading Volume". In: *The Journal of Finance* 55.5, pp. 2017–2069. DOI: https://doi.org/10.1111/0022-1082.00280. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/0022-1082.00280. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/0022-1082.00280.

Leung, M. T., A.-S. Chen, and H. Daouk (2000). "Forecasting exchange rates using general regression neural networks". In: *Computers & Operations Research* 27.11-12, pp. 1093–1110.

Mäkinen, Y. et al. (2019). "Forecasting jump arrivals in stock prices: new attention-based network architecture using limit order book data". In: *Quantitative Finance* 19.12, pp. 2033–2050.

Nousi, P. et al. (2019). "Machine learning for forecasting mid-price movements using limit order book data". In: *Ieee Access* 7, pp. 64722–64736.

Ntakaris, A. et al. (2018). "Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods". In: *Journal of Forecasting* 37.8, pp. 852–866.

Ntakaris, A. et al. (2019). "Feature engineering for mid-price prediction with deep learning". In: *Ieee Access* 7, pp. 82390–82412.

Passalis, N. et al. (2018). "Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 4.6, pp. 774–785.

Rapach, D. E., J. K. Strauss, and G. Zhou (2013). "International stock return predictability: what is the role of the United States?" In: *The Journal of Finance* 68.4, pp. 1633–1662.

Rouwenhorst, K. G. (1998). "International momentum strategies". In: *The journal of finance* 53.1, pp. 267–284.

Ruck, D. et al. (1990). "Feature Selection Using a Multilayer Perceptron". In: *Journal of Neural Network Computing* 2.2, pp. 40–48.

Scikit-learn (2021). "1.1. Linear Models — scikit-learn 0.24.1 documentation. [online] Scikit-learn.org." In: *https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (Accessed 28 February 2021).*

Sharpe, W. F. (1964). "Capital asset prices: A theory of market equilibrium under conditions of risk". In: *The journal of finance* 19.3, pp. 425–442.

Srivastava, N. et al. (2014). “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Tieleman, T. and G. Hinton (2012). “Lecture 6.5-RMSProp, COURSERA: Neural networks for machine learning”. In: *University of Toronto, Technical Report*.

Tran, D. T. et al. (2019). “Temporal Attention-Augmented Bilinear Network for Financial Time-Series Data Analysis”. In: *IEEE Transactions on Neural Networks and Learning Systems* 30.5, pp. 1407–1418. DOI: 10.1109/TNNLS.2018.2869225.

Wang, Y.-H. (2009). “Nonlinear neural network forecasting model for stock index option price: Hybrid GJR–GARCH approach”. In: *Expert Systems with Applications* 36.1, pp. 564–570.

Yao, J., Y. Li, and C. L. Tan (2000). “Option price forecasting using neural networks”. In: *Omega* 28.4, pp. 455–466.

Zhang, L. (2005). “The Value Premium”. In: *The Journal of Finance* 60.1, pp. 67–103. DOI: https://doi.org/10.1111/j.1540-6261.2005.00725.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.2005.00725.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2005.00725.x.

APPENDIX A. Neural Networks

This Appendix holds figures presenting structures for different neural networks used in this study.

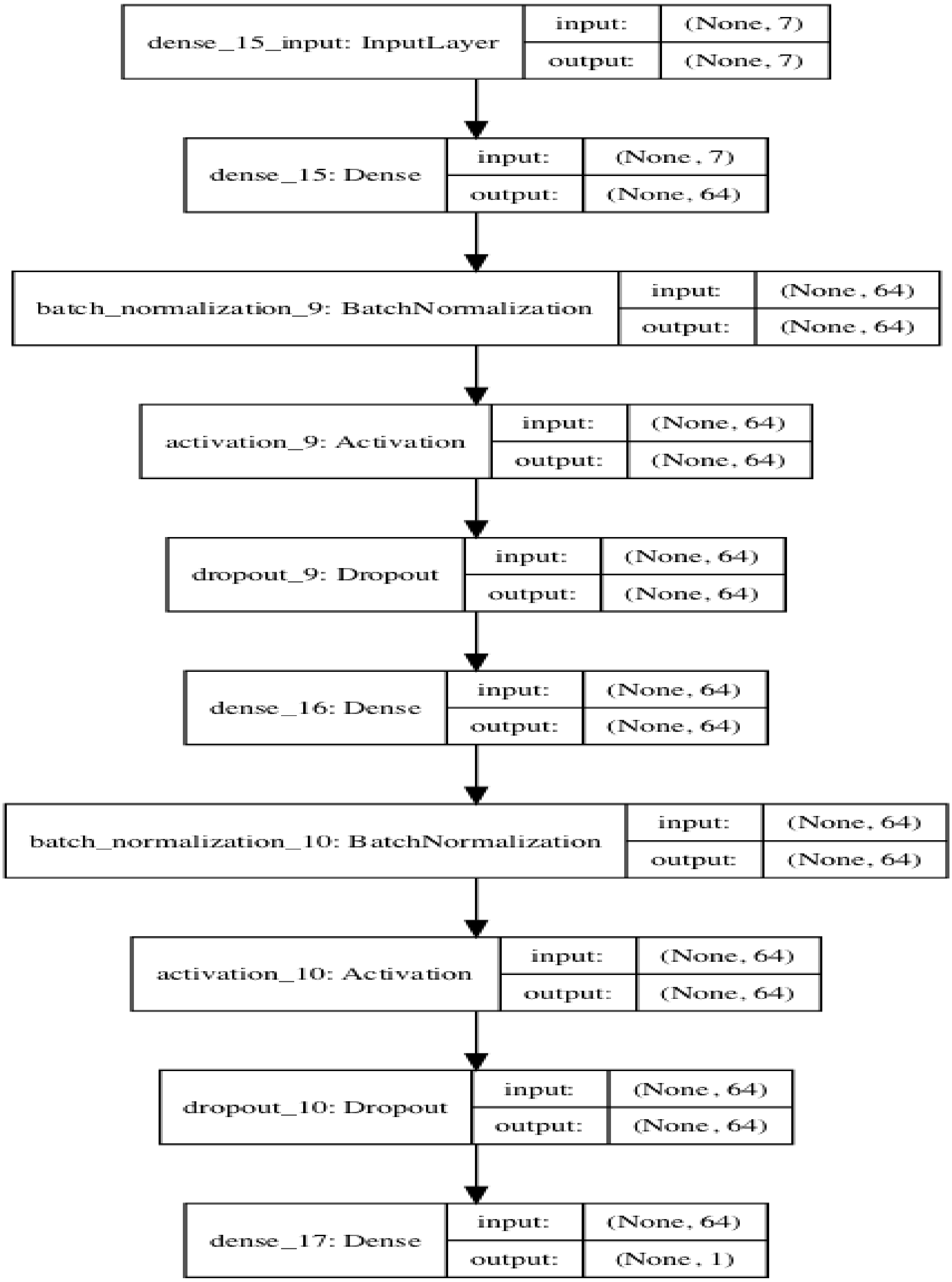

Figure 1 *Figure presents neural network structure for MLP-64 model.*

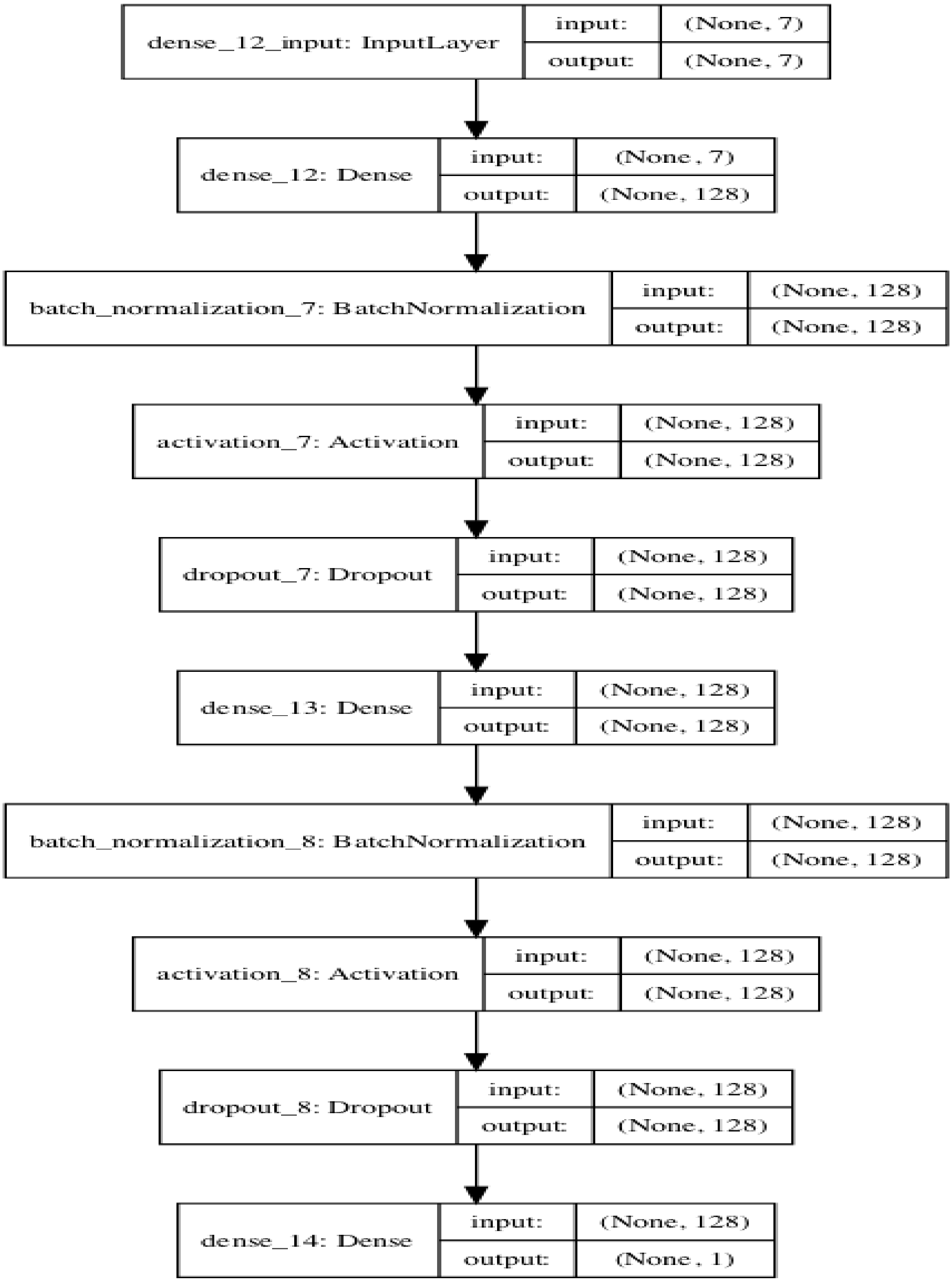

Figure *2 Figure presents neural network structure for MLP-128 model.*

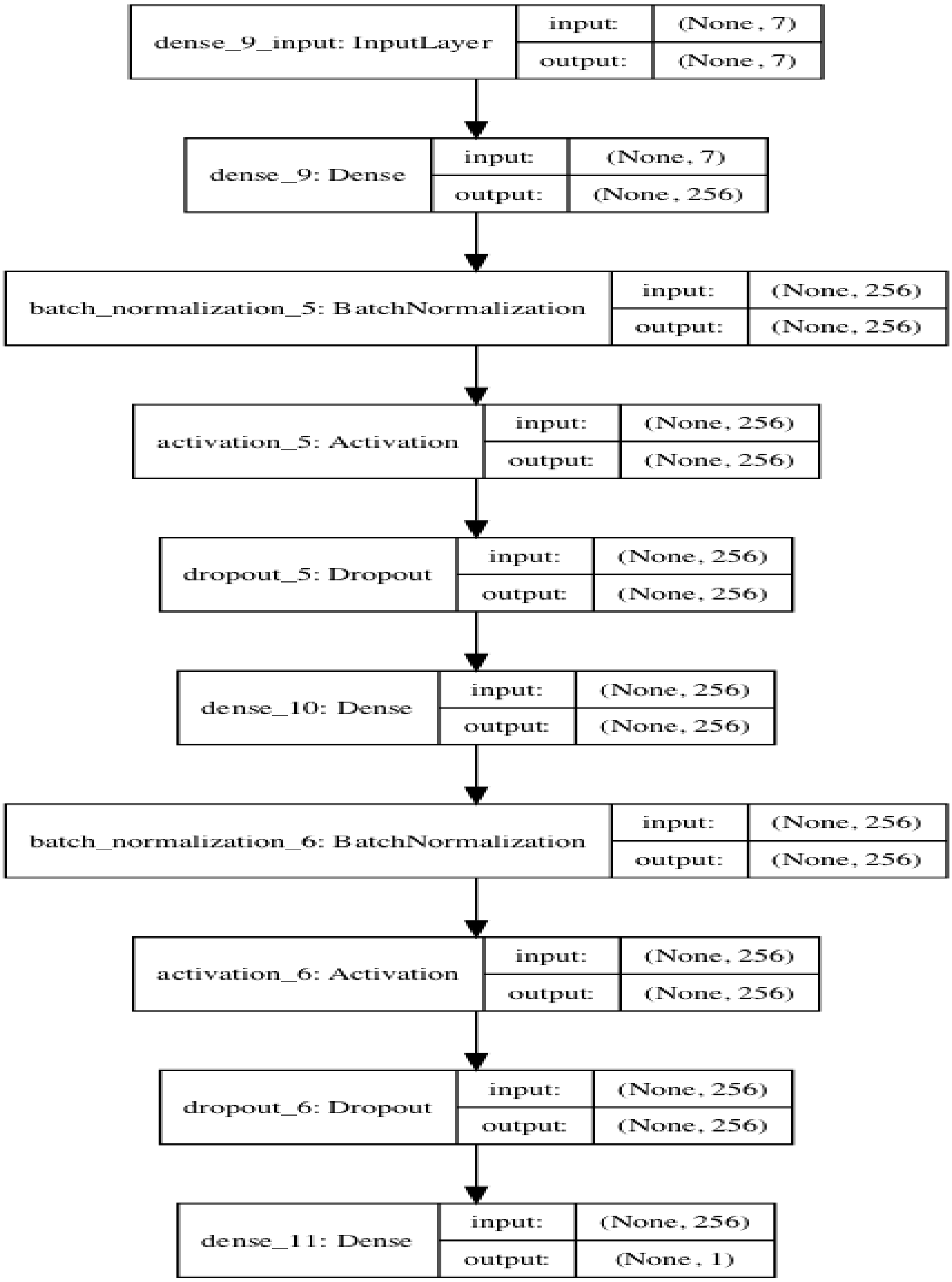

Figure 3 *Figure presents neural network structure for MLP-256 model.*

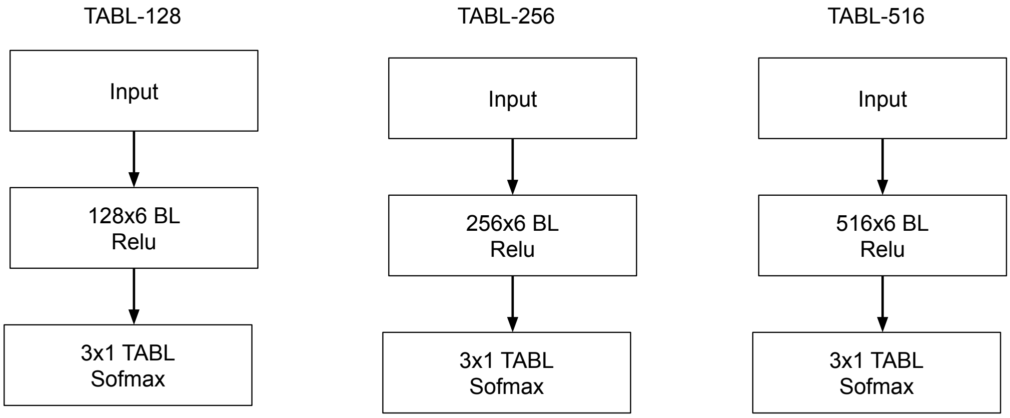

Figure 4 *Figure presents neural network structure for three TABL networks used in this study. Left on the figure is TABL-128 model. Middle in the figure is the TABL-256 and on the right is the TABL-516 model.*

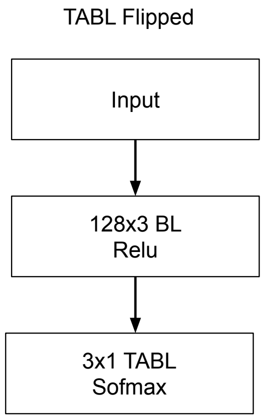

Figure 5 *Figure presents neural network structure for TABL Flipped model.*

APPENDIX B. Supporting results

This appendix holds supporting results for this study.

Table 1 *Summary statistics for raw data used in this thesis. Market capitalisation, mcap in table, below and price-to-book ratio, pb in table, are values obtained straight from the original data. Returns are calculated from the adjusted prices. Returns are logarithmic returns and correspond to momentum indicators used in this thesis and to prediction target used in this thesis. Risk free rate is not reduced from these returns. This raw data contains 11 498 786 observations. Original data has more observation but in this data, which is the final data for this thesis, missing values for data items in table are removed from the data. This removing missing data explains why next 21d returns differ from past 21d returns. When missing values are not removed these items have same statistics.*

Data item	Mean	Std. Deviation	Median	25%	75%
pb	3.41	878	1.70	0.900	3.30
mcap	6500	74300	520	78.4	2810
Next 21d Return	-0.00610	0.176	$5.30 * 10^{-4}$	-0.0610	0.0597
Prev 5d Return	$-5.43 * 10^{-4}$	0.0931	0.000	-0.0273	0.0270
Prev 21d Return	-0.00333	0.170	$9.31 * 10^{-4}$	-0.0606	0.0606
Prev 63d Return	-0.0125	0.283	0.00628	-0.112	0.111
Prev 126d Return	-0.0239	0.400	0.0139	-0.163	0.162
Prev 252d Return	-0.0300	0.719	0.0719	-0.270	0.336

Table 2 *$R^2_{Comparison}$-statistics for model compared to another model as percentage. In table column model is compared to row model. Above zero means column model is more accurate model and below zero means column model is less accurate than row model*

Model	OLS-All	OLS-Standard	MLP-64	MLP-128	MLP-256
OLS-All		-0.35%	1.00%	0.95%	1.17%
OLS-Standard			1.35%	1.30%	1.52%
MLP-64				-0.05%	0.17%
MLP-128					0.22%

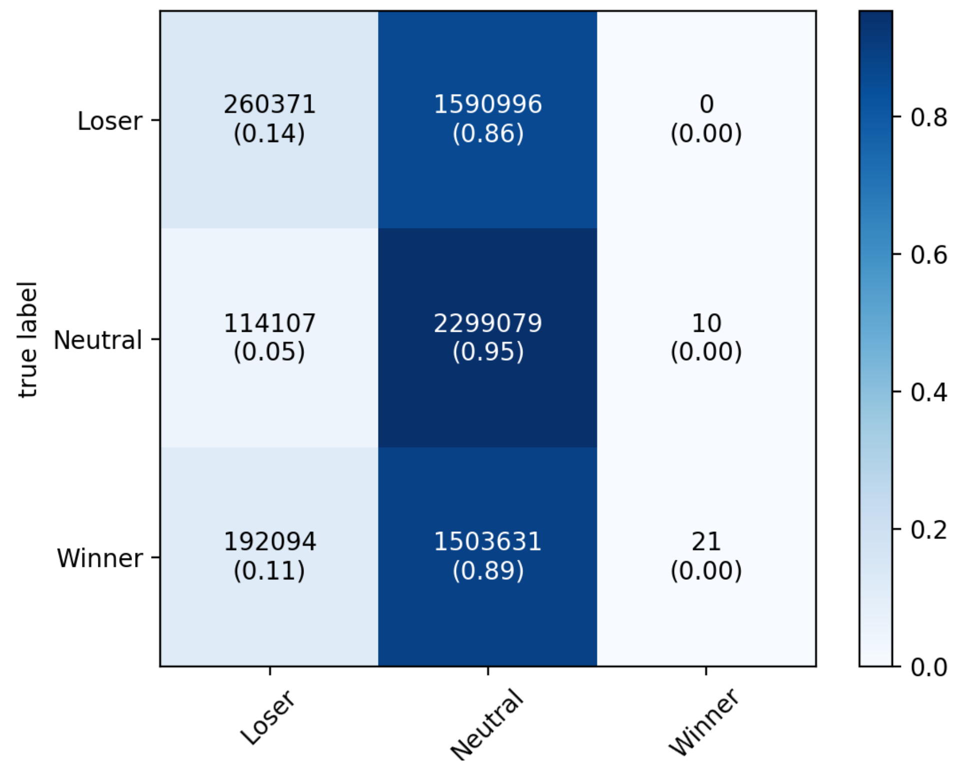

Figure 6 *Confusion matrix for logistic regression model. Numbers are real confusions and parentheses hold normalized confusion.*

Table 3 *Summary statistics for MLP portfolios' monthly excess returns.*

Metric	All	Positive MLP	Negative MLP
Mean	0.46%	0.65%	0.17%
Std. Deviation	4.2%	3.5%	5.4%
Median	0.64%	0.93%	0.30%
25%	-1.9%	-1.2%	-3.1%
75%	3.0%	2.7%	3.6%
Max	18%	16%	20%
Min	-15%	-12%	-17%

Table 4 *Summary statistics for TABL portfolios' monthly excess returns.*

Metric	All	TABL Losers	TABL Neutral	TABL Winner
Mean	0.47%	0.1%	0.53%	0.51%
Std. Deviation	4.2%	5.6%	3.0%	7.0%
Median	0.61%	0.31%	0.79%	0.22%
25%	-1.9%	-3.3%	-0.92%	-3.1%
75%	3.0%	3.6%	2.3%	3.7%
Max	18%	23%	10%	82%
Min	-15%	-20%	-12%	-21%

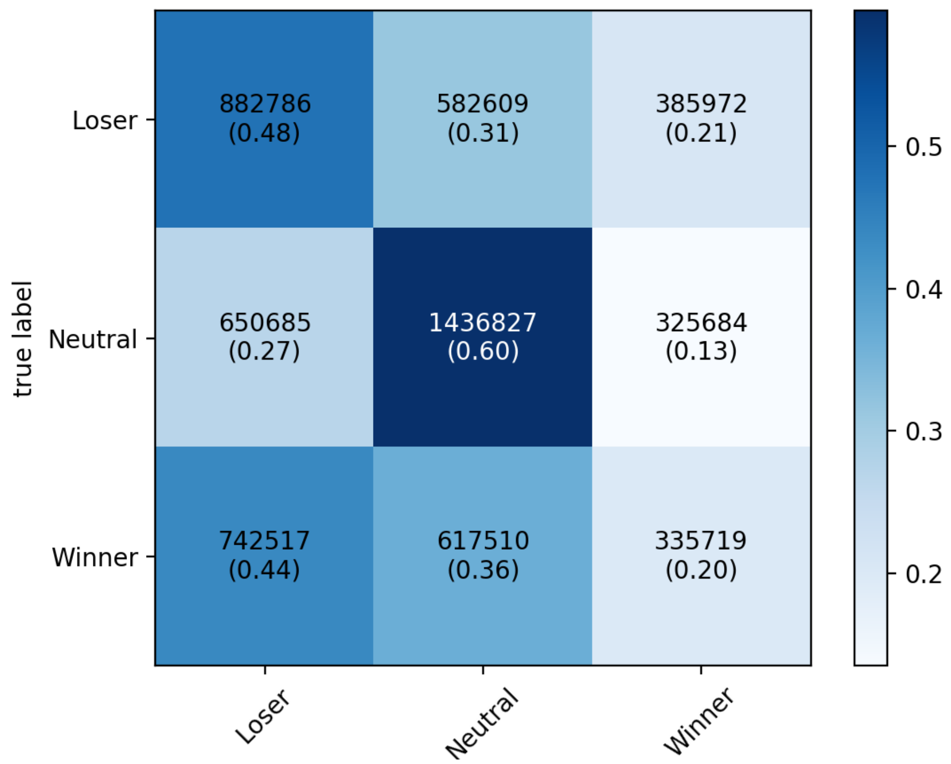

Figure 7 *Confusion matrix for TABL-128-model. Numbers are real confusions and parentheses show normalised confusion.*

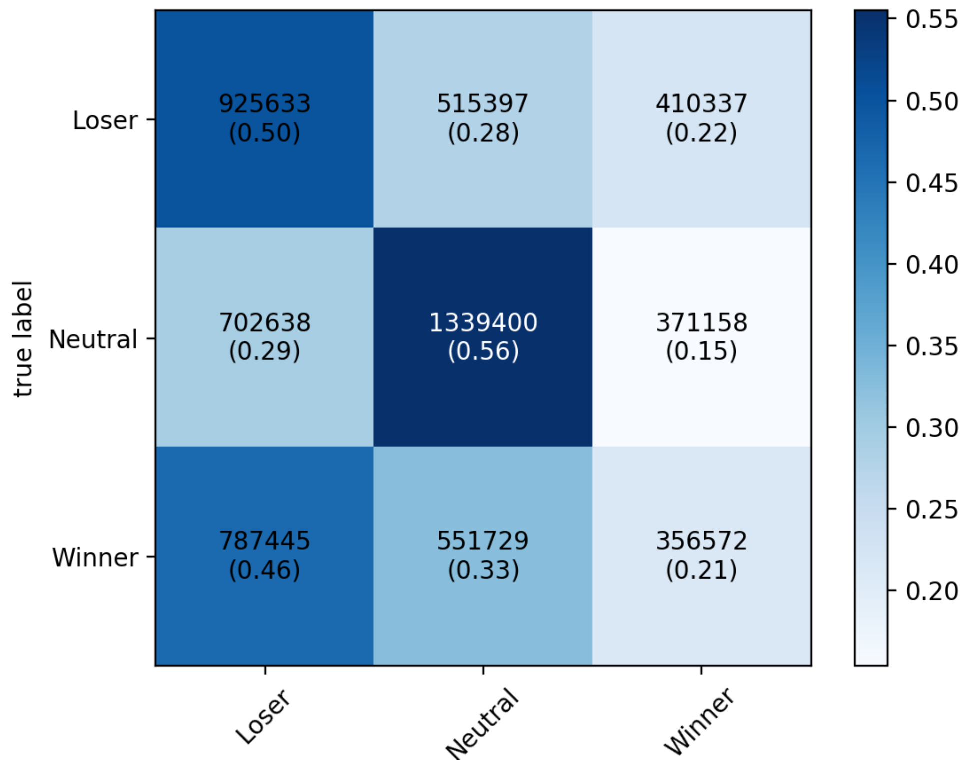

Figure 8 *Confusion matrix for TABL-256 model. Numbers are real confusions and parentheses hold normalised confusion.*

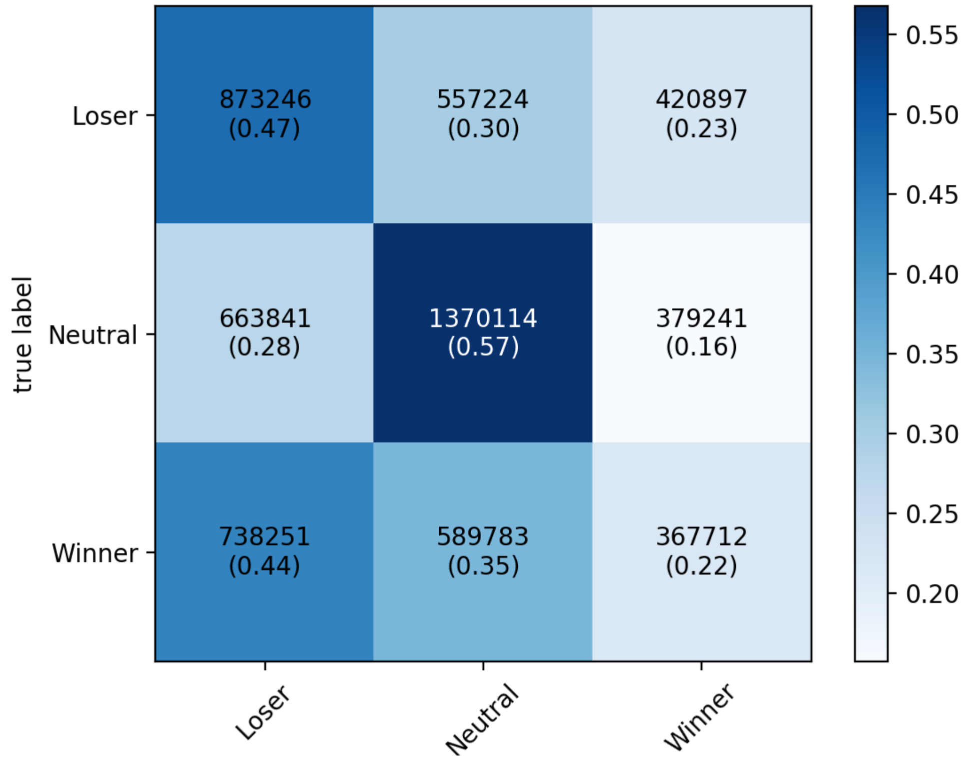

Figure 9 *Confusion matrix for TABL 516 model. Numbers are real confusions and parentheses contain normalised confusion.*

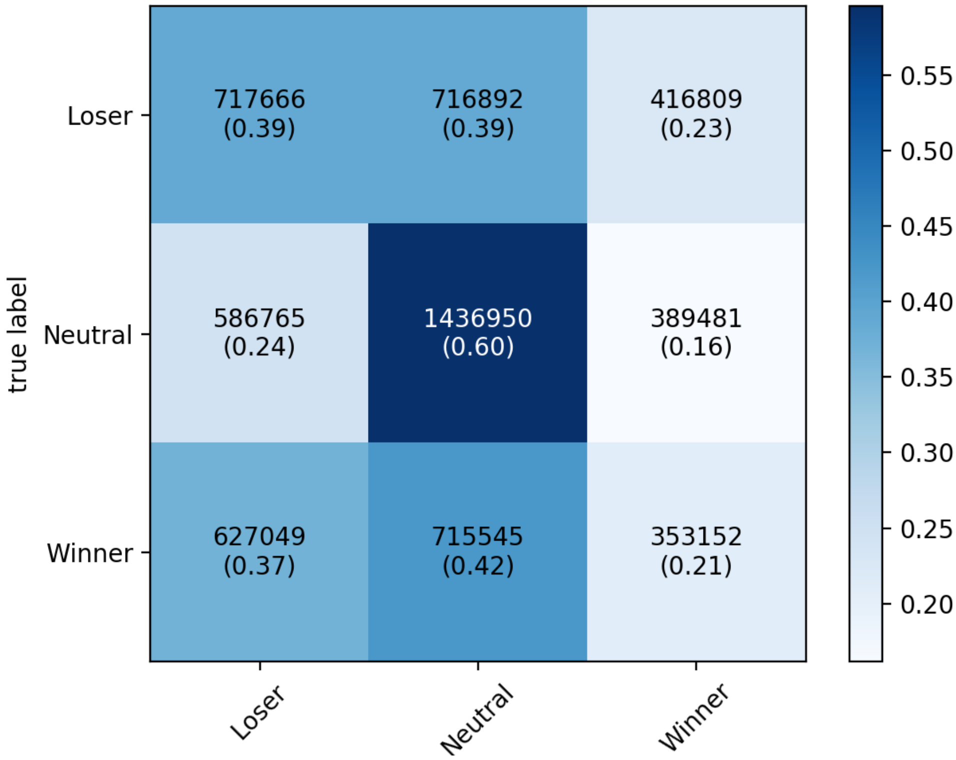

Figure 10 *Confusion matrix for TABL Flipped model. Numbers are real confusions and parentheses contain normalised confusion.*