©Copyright 2022
Yuming Liu

Essays on Empirical Asset Pricing and Time Series Forecasting

Yuming Liu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Eric Zivot, Chair
Chang-Jin Kim
Jing Tao

Program Authorized to Offer Degree:
Economics

University of Washington

Abstract

Essays on Empirical Asset Pricing and Time Series Forecasting

Yuming Liu

Chair of the Supervisory Committee:
Eric Zivot
Department of Economics

In this dissertation, I aim to forecast U.S. stock returns via state-space and probabilistic deep learning models.

In the first chapter, I propose new state-space models with stock and accounting variables to estimate the expected market returns. These approaches uncover the information existing in unobserved state variables through the predictive updating system based on the Kalman filter technique. The one-step-forward in-sample prediction for the state-space model with stock variables has R^2 as 13%, where the expected market return has a persistent component. I further improve the performance of forecasting market returns by incorporating accounting variables with a state-space model, having a higher R^2, 18%. Both expected market returns and expected returns on equity have persistent components, but expected returns on equity are more persistent than expected market returns. Results from out-of-sample predictions further reinforce the forecastability of market returns based on proposed models, especially for short-range predictions.

In the second chapter, I conduct a comparative analysis of advanced deep-learning models for forecasting U.S. market returns. These approaches uncover more information existing in the same dataset from the first chapter. I present a higher out-of-sample R^2, ranging from 42.33% to 65.80%, compared with classical time series models. By introducing the family of probabilistic deep-learning models, I reinforce the argument that accounting variables are

more informative than stock variables for predicting market returns. Also, I build a link between traditional metrics and advanced neural nets by having an extension of state-space models, the deep state-space model. Innovative deep-learning models simplify estimations of multiple time series and highlight the value of neural nets without losing interpretability.

In the third chapter, I conduct an analysis of forecasting the U.S. stock returns based on the probabilistic deep learning methods described in the second chapter. By estimating aggregate-level stock returns, I find that DF-RNN and DeepVAR provide the most accurate results, with out-of-sample R^2 ranging from 63.14% to 76.51%. DSSM precisely estimates firm-level annual stock returns, with a 4.52% R^2 on the testing dataset. Also, by building a zero-net-investment trading strategy, I find that DeepAR and DSSM can help to construct profitable portfolios with cumulative returns ranging from 4.26% to 5.13% on out-of-sample periods. As a result, probabilistic deep learning models can generate state-of-the-art predictions of U.S. stock returns at both aggregate and firm levels.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to the chair of my supervisory committee, Professor Eric Zivot, for his constant support and insightful guidance throughout my doctoral study. His invaluable advice and patience navigate me through my research and teaching journey. Additionally, this endeavor would not have been possible without his financial support.

I am also deeply indebted to my committee members. I received generous support and encouragement from Professor Chang-Jin Kim, who also inspired me to pursue topics in time series. His course and research enlightened me with plenty of ideas and resources. Professor Thomas Gilbert's continuous support and feedback helped me to acquire the data and profoundly expand my vision with new ideas. Professor Jing Tao consistently helped me to understand econometrics and intuitions with her critical and insightful comments. I am also grateful to my GSR member, Professor Simon Du, for his fruitful discussions and kindness. My appreciation extends to Chloe Mahar, Simon Reeve-Parker, Heidi Hannah, Kim Lee, Ahna Kotila, and Chris Fendrich for the administrative support. The completion of the dissertation would not have been possible without their involvement.

Finally, I am particularly thankful to my family and friends for their unconditional love and support. With their tremendous understanding and encouragement, I can pursue my academic goal and keep being passionate about my work.

DEDICATION

To my partner and parents,
for their love,
and trust.

Chapter 1

PREDICTING THE U.S. MARKET RETURNS VIA STATE-SPACE MODELS

1.1 Introduction

Starting from the 1920s, economists have been forecasting stock market returns or equity premiums with different methods or data. Though Lettau and Ludvigson (2001) [57] mention that excess returns are predictable based on dividends or earnings, there are contradictory findings regarding predicting stock returns. Researchers further test return predictions on different variables based on various algorithms.

In this paper, I utilize both aggregate stock and accounting variables to examine the annual stock return forecastability of various models, such as vector autoregressive model (VAR) and state-space model (SSM). By forming latent variables' processes, I estimate the annual expected market returns using the conditional expected stock and accounting variables to create unobserved time-series processes. Then, I combine the latent system with log linearized market returns to derive the predictive regressions estimated through the Kalman filter based on likelihoods. I find that more information is captured from expected state variables through the latent processes, which can improve the performance of models. These models give better predictions of market returns, with R^2 ranging from 13% to 18%.

I also compare the latent processes with other common methods. I first test basic linear regressions based on different variables and extend the simple model to an VAR model to simulate the process by including relations over time. The R^2 ranges from 6% to 10% for these models.

For state-space models, I start with defining latent variables, expected dividends, and expected returns. Following Cochrane (2007 & 2008) [22] [23] and Campbell and Shiller's decomposition (Campbell & Shiller, 1988a) [16], I generate a time-varying process for latent stock or accounting variables. The state-space model with time-varying latent stock variables improves the forecastability of aggregate stock returns. The stability of the state-space model is further tested based on the training and testing datasets.

Following Vuolteenaho (2000) [90], I make three assumptions for the analysis in this paper. First, book and market value are assumed to be positive, which is fundamental for the clean surplus identity (Equation (1.22)). Second, the dividend-price ratio and the book-to-market ratio are assumed to be stationary. This assumption helps to specify a value around which the price changes. The third assumption is the clean surplus identity, which links the current book value with earnings, dividends, and the last period's book value.

My main contributions are the following. First, I examine the traditional market return forecasting models on new datasets. I find that the results from VAR are the best, but they cannot capture the potential movements in the latent variables. By examining the residuals from VAR models, I believe there is some critical information in residuals that the model does not capture. Thus, it is crucial to introduce the state-space model to uncover the dynamics behind observed variables further.

Second, I improve the forecastability of aggregate stock market returns by using two new state-space models. The first state-space model is based on dividend yield and includes the market return in the measurement equation directly rather than calculates the returns based on predicted dividend growth rates and dividend yields (Binsbergen & Koijen, 2010, [88]). Then, I can bring more information existing in unobserved variables to the predictive system, which highly improves the predictability. And, expected market returns are more persistent than expected dividend growth rates.

My second state-space model is based on aggregate accounting data and generates the best forecasting results. Due to the instability in dividend policies over time, I use the book-to-market ratio as a substitute for the dividend-price ratio and create the structure based on the clean surplus identity (Equation (1.22)) and first-order Taylor expansions. With accounting variables, I find that market returns are consistently predicted with relatively significant improvements, where expected returns on equity are more persistent than expected market returns.

By decomposing variance of unexpected market returns, I find that cash-flow news accounts for 3.7% or 147.9% of the variance, discount news accounts for 135.8% or 9.5% of the variance, and the covariance between these two news accounts for the rest. The decomposition suggests that accounting data can help to significantly explain the variation of unexpected market returns.

Further, both state-space models show great forecastability of short-range market returns because of the persistent state variables over time. Considering the unstable dividend policy, the accounting-based state-space model performs better than the other due to the persistent of cash-flow state variables, expected returns on equity.

The organization of this paper is as follows. In Section 1.2, I review the background of return forecastability, including historical forecasting approaches and previous state-space models. In Section 1.3, I describe the stock and accounting data for this paper. In Section 1.4, I discuss two state-space models with different information sets, and in Section 1.5 I outline the results from different methods. Section 1.6 contains my conclusions.

1.2 Literature Review

There is a long research history of forecasting stock returns. Starting from the 1920s, Dow and Selden (1920) [27] try to optimize the trading algorithms based on forecasting stock returns using dividend ratios. Different papers use different algorithms or variables to forecast the excess returns, considering various measurements of stock risks.

1.2.1 Background of Forecasting

For forecasting stock returns, the dividend-price ratio is used by Campbell (1987) [13], Campbell and Shiller (1988a) [16], Fama and French(1988) [28], Hodrick (1992) [47], Wolf (2000) [93], Lewellen (2004) [58], Campbell and Yogo (2006) [15], Ang and Bekaert(2006) [2], Cochrane(2007) [22], Goyal and Welch (2008) [37], and Van Binsbergen and Koijen (2010)

[88]. Most of these authors argue that the dividend yield can be used to forecast stock returns, although the strength of predictions varies considerably across studies.

The research studying the dividend-price ratio contains contradictory findings regarding the forecastability of excess stock returns. For example, Goyal and Welch (2008) [37] investigate the forecasting power of a diverse group of stock and accounting variables based on out-of-sample observations. They find that the prediction models change significantly over time and that most of the predictors perform worse than predicting returns using the historical means. They also point out that available information cannot help investors make additional profits, Cochrane (2007) [22], however, argues that Goyal and Welch's results only show difficulties in using predictions to form trading strategies. In other words, the out-of-sample R-square is a statistic that evaluates the performance or usefulness of making market decisions based on the prediction, which does not indicate the rejection of the forecastability of stock returns. Cochrane (2007) tests the hypothesis that "if returns are not predictable, then dividend growth must be predictable." Based on the absence of dividend growth predictability, it can indirectly defend the forecastability of stock returns.

Other researchers also estimate stock returns based on accounting variables. They find that forecasting using earning price ratio performs better than forecasting using dividend-price ratios, which is initially tested in a VAR model by Campbell and Shiller (1988b) [17] and Lamont (1998) [56]. Currently, researchers proved a positive relationship between firm-level earnings and stock prices or returns (Choi et al. (2016) [20], Bonsall et al. (2013) [10]). However, for the aggregate-level market, there is a negative relationship between earnings and stock market returns (Kothari, Lewellen, and Warner (2006) [54], Sadka (2007) [80], Sadka and Sadka (2009) [81], Hirshleifer, Hou, and Teoh (2009) [45], Patatoukas (2013) [70]). They also examine the relation between earnings news and stock returns and conclude that aggregate returns are forecastable.

1.2.2 Background of Variance Decomposition

Campbell and Shiller (1988b) [17] decompose unexpected stock returns into two parts, changes in expected future cash flow (dividends) news and changes in expected future discount rate (stock returns) news. Later, Campbell (1991) [14] shows that the variance of future cash flow news accounts for one third of the variance of unexpected aggregate stock returns.

Motivated by Modigliani-Miller's dividend irrelevant theory, the instability in aggregate dividend policy, and the weak results regarding forecasting long-horizon stock returns using dividend price ratios, Vuolteenaho (2000) [90] develops a return model which builds upon the relationships among book-to-market ratio, return on equity (ROE), interest rates, and returns. Using this model, Vuolteenaho (2002) [91] finds that cash-flow news significantly drives the firm-level stock returns, while expected-return information is significantly driven by aggregate-level components.

Then, the current period stock return can be separated into three parts: change in conditional expected returns, change in expected cash-flow news, and change in expected-return news. While people estimate the firm-level relations between returns and earnings, they conclude that $Cov(N_{cf,i,t}, \Delta X_{i,t}) > 0$ and $Cov(R_{i,t}, \Delta X_{i,t}) > 0$, where $\Delta X_{i,t}$ denotes the changes in earnings for firm i. Hecht and Vuolteenaho (2006) [44] further extend the relations to the remaining variables in the decomposition, stating that the change in earnings is correlated with lagged and leading information. Moreover, the aggregate earning is negatively correlated with the return news.

1.2.3 Background of Estimated Models

When forecasting stock returns using dividend price ratios and other accounting data, the type of forecasting model is important. In empirical work, the Vector Autoregressive model (VAR) is the most common model used for predicting stock returns, see examples such as Cochrane (2007) [22] and Brandt and Kang (2004) [11]. In contrast to the VAR model, Rytchkov (2007) [79], Pastor and Stambaugh (2009) [69], Van Binsbergen and Koijen (2010) [88], and Monache et al. (2021) [64] focus on predicting stock returns based on the state-space model and treating the expected stock return as a latent variable.

State-space models allow time variation in parameters and automatically apply restrictions in the updating process of latent components in the model. For modeling returns based on present-value representations, state-space models can handle complex relations and efficiently utilize all the information. After considering the movements in market-wide data, all the papers above show evidence regarding improvement in stock return forecastability.

Besides the dividends and accounting variables, improvements in computational technologies induce lots of research based on event-driven data or different markets. These research focus more on interesting predictors, feature selections, and data mining. For example, Bijl et al. (2016) [9] find that data from Google traffic can predict the stock returns. Salisu et al. (2019) [84] proposed an alternative approach to forecast market returns based on Bitcoin prices. In other words, the additional available public or private information further improves predictability.

There are also other interesting and useful factors, including investor sentiments (Huang et al. (2015) [50], Li et al. (2015) [60]), Ren et al. (2018) [76], Audrino et al. (2020) [5]), financial news (Atkins et al. (2018) [4], Nam and Seong (2019) [67]), technical indicators (Neely et al. (2014) [68], Lin (2018) [61], Dai et al. (2020) [25]), and others. Also, more advanced and maturer machine learning techniques are widely used to improve the predictability of market returns, including big data with principal component analysis (De Mol et al. (2008) [26], Brodie et al. (2009) [12], Carrasco and Rossi (2016) [18], Reichlin et al. (2017) [75]), regression trees (Rossi (2018) [77], Rasekhschaffe (2019) [73]), deep learning (Chong et al. (2017) [21], Fischer and Krauss (2018) [32], Hu et al. (2018) [49]), etc. In this paper, I focus on building state-space models based on stock and accounting variables, and the modern methods are revisited in the next paper.

1.3 Data

Following Sadka and Sadka (2009) [81] and Vuolteenaho (2002) [91], for the analysis in this paper, I generate aggregate-level dividend growth rates, dividend-price ratios, returns, returns on equity, book equity, market value, and book-to-market ratio.

1.3.1 Basic Data

The basic data contain all firm data in the Center for Research in Security Prices (CRSP) and COMPUSTAT databases obtained from Wharton Research Data Services (WRDS) [85]. The CRSP data contains with-dividend and without-dividend monthly stock market returns based on the value-weighted portfolios of NYSE, AMEX, and NASDAQ stocks for the period 1950 - 2019. The COMPUSTAT annual research file contains the relevant accounting information for most publicly traded firms, including book values, market values, and returns on equity etc. In this paper, I additionally use rolled-over 90-day Treasury-bills (the risk-free rate) and the Consumer Price Index (CPI) from CRSP and compute the corresponding series for dividend growth rates, dividend-price ratios, returns, returns on equity, and risk-free rates. All variables are at the annual frequency.

1.3.2 Data Manipulations

By utilizing the value-weighted stock market returns from CRSP, I generate the price-dividend ratio and dividend growth ratio. For firm-level data from COMPUSTAT, firms must have December as the fiscal-year end to align accounting variables across firms. To filter out data errors, I exclude firms with less than $10 million market values and more than 100 or less than 0.01 book-to-market ratio.

Firm-level variables are calculated as follows. The market value of equity is the product of common outstanding shares and the closing price in certain fiscal years. For book equity, I use the total common equity (data item 60). If the data is not available, I use the liquidity value (data item 235) as a substitute. Considering taxes, short-term and/or long-term deferred taxes (data item 35 and 71) are added to book equity if available. If neither the total common equity nor the liquidity value is available, I use the clean surplus identity (Equation (1.22)) to approximate book value. All firm-level book equity must be non-negative to be included in the analysis.

Firms' net incomes (data item 172) are regarded as earnings, and if the data is missing, earnings are approximated using the clean surplus identity (Equation (1.22)). Return on equity (ROE) or profitability is the earnings over the last period's book equity. Intuitively, firms cannot lose more than their book values. Thus, if firms have negative earnings, the absolute value of earnings must be smaller than its book equity.

To convert firm-level data into aggregate-level data, I utilize market capitalization to calculate value-weighted variables. Market-level data are calculated as the value-weighted mean of existing variables scaled by the price level in the fiscal year.

1.3.3 Descriptive Statistics

Table 1.1 shows the descriptive statistics for the aggregate-level variables, including means, standard deviations, and quantiles for log excess returns, log excess return on equity, and other variables from 1950 to 2019. These statistics are similar to the results from Vuolteenaho (2002) [91] and Hecht and Vuolteenaho (2006) [44], though the length of data is different. Comparing log returns with the log excess return on equity, the volatility in the stock market return is larger than the accounting-based ROE (standard deviation of 0.16 vs. 0.05). Also, the log dividend growth rate contains great fluctuations over time as the standard deviation is 5 times larger than its mean.

Comparing the log dividend-price ratio with the log book-to-market ratio, there are similar standard deviations or volatility. This can be evidence for the following state-space model, in which the book-to-market ratio can be used as a substitute for the dividend-price ratio. And the log excess earnings can be the substitute for dividend growth rates, considering the similar quantiles in the sample data (except the extreme values).

Table 1.2 shows the correlations among the logged variables. Similar to previous liter-

Table 1.1: Descriptive Statistics: Summary

	Mean	SD	Min	25%	Median	75%	Max
r - r_f	0.0576	0.1628	-0.4959	-0.0358	0.0944	0.1638	0.3970
Δd	0.0208	0.1172	-0.2747	-0.0614	0.0106	0.0798	0.3268
dp	-3.4429	0.3417	-4.1785	-3.6771	-3.4735	-3.1795	-2.8394
e - r_f	0.1024	0.0524	0.0011	0.0654	0.0935	0.1405	0.2794
θ	-0.4452	0.3151	-1.1664	-0.6735	-0.4788	-0.2110	0.1760

This table reports means, standard deviations, and quantiles of log excess return, r - r_f, log dividend growth rate, Δd, log dividend-price ratio, dp, log excess return on equity, e - r_f, and log book-to-market ratio, θ.
These statistics are estimated based on the data from CRSP and COMPUSTAT from 1950 - 2019.

ature, such as Cochrane (2007) [22], the correlation between the log dividend growth rate and log excess return is about 0.7, which shows that the log dividend growth rate shares some common patterns with the log returns. There is a similar relationship between the log dividend yield and the log book-to-market ratio, which further reinforces the use of the clean surplus identity.

The log dividend-price ratio has a small correlation with market returns, but large correlations with accounting returns. This suggests that the clean surplus identity can support a model with better performance for forecasting market returns, as the variables are more related to the aggregate firm-level states.

The negative correlation (-0.6) between log excess return on equity and log dividend yield shows evidence that firms' returns on equity are reduced while distributing an increasing number of dividends. There is also similar relation between the log book-to-market ratio and the log excess return on equity, given the correlation as -0.5.

One notable feature of the data is that the correlation between log excess return on equity and log returns is about 0.11, which illustrates the small relationship between market and accounting returns. Intuitively, in an open economy, considering all the firms as a whole, aggregate returns on equity is not significantly affected by the domestic aggregate-level returns.

Table 1.2: Descriptive Statistics: Correlations

	r - r_f	Δd	dp	$e - r_f$	θ
r - r_f	1				
Δd	0.6942	1			
dp	-0.0028	0.0005	1		
$e - r_f$	0.1127	0.1891	-0.6413	1	
θ	-0.1637	-0.1154	0.8965	-0.4871	1

This table reports correlations among log excess return, r - r_f, log dividend growth rate, dg, log dividend-price ratio, dp, log excess return on equity, e - r_f, and log book-to-market ratio, θ. Sample from 1950 - 2019, CRSP and COMPUSTAT.

1.3.4 OLS Forecasting Performance

This section summarizes the in-sample forecasting ability of stock and accounting variables for stock returns. Table 1.3 presents the regressions of real return and profitability on lagged dividend-price or book-to-market ratio as below.

$$\begin{aligned} r_{t+1} &= a + b dp_t + \varepsilon_{t+1} \\ r_{t+1} - r_{f,t+1} &= a + b dp_t + \varepsilon_{t+1} \\ \Delta d_{t+1} &= a + b dp_t + \varepsilon_{t+1} \\ r_{t+1} &= a + b\theta_t + \varepsilon_{t+1} \\ r_{t+1} - r_{f,t+1} &= a + b\theta_t + \varepsilon_{t+1} \\ e_{t+1} - r_{f,t+1} &= a + b\theta_t + \varepsilon_{t+1} \end{aligned} \tag{1.1}$$

where a denotes the intercept, b denotes the coefficient, r_{t+1} denotes the real log market return at time $t+1$, $r_{f,t+1}$ denotes the log risk-free rate at time $t+1$, Δd_{t+1} denotes the log dividend growth rate at time $t+1$, dp_t denotes the log dividend-price ratio at time t, θ_t denotes the log book-to-market ratio at time t, $e_{t+1} - r_{f,t+1}$ denotes the log excess return on equity at time $t+1$.

Table 1.3: OLS Forecasting Performance

Regressions	a		b		t	R^2	$\sigma(bx)$
	est	se	est	se			
$r_{t+1} = a + bdp_t + \varepsilon_{t+1}$	0.4808	0.1343	0.1200	0.0400	2.9988	0.0618	0.0412
$r_{t+1} - r_{f,t+1} = a + bdp_t + \varepsilon_{t+1}$	0.4041	0.1343	0.1013	0.0392	2.5816	0.0452	0.0347
$\Delta d_{t+1} = a + bdp_t + \varepsilon_{t+1}$	0.0605	0.0722	0.0109	0.0218	0.5012	0.0010	0.0037
$r_{t+1} = a + b\theta_t + \varepsilon_{t+1}$	0.1234	0.0252	0.1255	0.0576	2.1776	0.0572	0.0396
$r_{t+1} - r_{f,t+1} = a + b\theta_t + \varepsilon_{t+1}$	0.1086	0.0268	0.1195	0.0545	2.1931	0.0534	0.0377
$e_{t+1} - r_{f,t+1} = a + b\theta_t + \varepsilon_{t+1}$	0.0676	0.0176	-0.0802	0.0294	-2.7228	0.2321	0.0253

This table reports linear regressions based on real variables. *est* represents estimated values of coefficients and *se* denotes the HAC standard errors of estimations. r is the real log market return, r_f is the log risk-free rate, r - r_f is the log excess return, Δd is the log dividend growth rate, e - r_f is the log excess return on equity, dp is the log dividend-price ratio, and θ is the log book-to-market ratio. Annual data, 1950 - 2019, from CRSP & COMPUSTAT. t reports the t-values for the coefficient, b, in each regression. $\sigma(bx)$ shows the standard deviations of the fitted value of the regression.

The coefficients show the forecastability of log excess returns based on dividend-price and book-to-market ratios, which is mentioned by Cochrane (2007) [22], Hecht and Vuolteenaho (2006) [44], etc. Also, considering the coefficients and R^2 from the third and last regressions, we can see that variation in market dividend yield do not forecast future dividend growths, but that variation in market book-to-market ratio forecasts future growth opportunities. This is evidence which briefly illustrates that using the accounting variables can improve the performance of forecasting market returns.

The estimated standard deviation of expected returns from the first and fourth regressions are similar, around 4%, which is much less than the standard deviation of market returns from the sample, 16%. For the fitted expected dividend growth rate, its standard deviation is about 0.3% which is much more smaller than the sample standard deviation of 12%. These results are similar to those from Cochrane (2007) [22]. The standard deviation of the log excess ROE captures about half of the volatility from the sample, which is another evidence that the accounting earning is a better predictor of market returns than the dividend growth rate.

1.4 Models

In this section, I define and evaluate different models based on both stock and accounting variables, including vector autoregressive and state-sapce models.

1.4.1 Vector Autoregressive Model

The Vector Autoregressive Model (VAR) is the most common model used for multivariate prediction of returns and accounting variables. In this section, I present two models to capture dynamic relationships among log returns, log dividend growth rate, and log dividend-price ratio, and dynamic relationships among log returns, log returns on equity, and log book-to-market ratio. These models interpret the sample data, and tests if the model is useful for predicting expected returns. The first VAR model based on stock variables has the form in Equation (1.2).

$$\begin{bmatrix} r_t \\ \Delta d_t \\ dp_t \end{bmatrix} = \begin{bmatrix} a_r \\ a_d \\ a_{dp} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} r_{t-1} \\ \Delta d_{t-1} \\ dp_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{t,r} \\ \varepsilon_{t,d} \\ \varepsilon_{t,dp} \end{bmatrix} \tag{1.2}$$

which can be represented as:

$$z_t = A_t + Bz_{t-1} + \varepsilon_t \tag{1.3}$$

The VAR for accounting variables is shown in Equation (1.4), which is similar to the above VAR model.

$$\begin{bmatrix} r_t \\ e_t - r_{f,t} \\ \theta_t \end{bmatrix} = \begin{bmatrix} a_r \\ a_e \\ a_\theta \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} r_{t-1} \\ e_{t-1} - r_{f,t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{t,r} \\ \varepsilon_{t,e} \\ \varepsilon_{t,\theta} \end{bmatrix} \tag{1.4}$$

In general, with one lag, there are twelve coefficients and six covariance terms that need to be estimated from the 210 annual observations. To estimate the order of VAR consistently, Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), and Hannan-Quinn Information Criterion (HQC) are used. The optimal selection of the orders for the accounting-variable VAR model (Equation 1.2) is one. Even though the optimal orders for the stock-variable VAR model (Equation 1.4) is two, the higher-order VAR only gives lower adjusted R^2 due to the less degree of freedom. Thus, this section shows a short VAR model with order one.

Table 1.4 presents the results for each regression in the estimated VAR models (Equation (1.2) & (1.4)), along with the results from serial correlation tests. Similar to Vuolteenaho (2002) [91], the expected returns on equity is high when the past return and past returns on equity are high, and the log book-to-market ratio is low. It also shows that the expected dividend-price ratio or the book-to-market ratio largely depends on its past ratio.

Further, the serial correlation test has a 0.0001 p-value for the stock-variable VAR model, indicating strong autocorrelations in the residuals. A stock-variable VAR model with two lags still shows the existence of serial correlations in the residuals with a p-value equals to 0.03. While the p-value for the accounting VAR model is about 0.2, which fails to reject the null hypothesis that there is no serial correlation in the residuals. This also illustrates that accounting variables can generate better forecasting results.

As a result, the log returns are poorly estimated based on the two models, but the dividend-price and book-to-market ratios are firmly estimated based on past ratios. Based on this result, I present a state-space model with a latent AR(1) process for the expected

Table 1.4: Short VAR for Aggregate Stock Market

	r_t	Δd_t	dp_t	r_t	$e_t - r_f$	θ_t
const	0.495	0.094	-0.269	0.104	0.016	-0.040
	(0.197)	(0.131)	(0.129)	(0.046)	(0.009)	(0.041)
r_{t-1}	-0.272	-0.315	-0.043	-0.036	0.002	0.011
	(-0.162)	(0.107)	(0.106)	(0.123)	(0.024)	(0.110)
Δd_{t-1}	0.389	-0.024	-0.429	-	-	-
	(0.230)	(0.152)	(0.150)	-	-	-
dp_{t-1}	0.121	0.014	0.922	-	-	-
	(0.057)	(0.037)	(0.037)	-	-	-
$e_{t-1} - r_f$	-	-	-	0.309	0.770	-0.341
	-	-	-	(0.448)	(0.085)	(0.401)
θ_{t-1}	-	-	-	0.146	-0.020	0.855
	-	-	-	(0.073)	(0.014)	(0.066)
R^2	0.109	0.214	0.908	0.0659	0.662	0.791
Adjusted R^2	0.067	0.177	0.903	0.0221	0.646	0.781
Serial Test (p-value)		0.0001			0.2	

This table reports the parameter estimates for the short VAR. This table includes the log stock market return (r), log dividend growth rate (Δd), log dividend-price ratio (dp), log excess return on equity ($e - r_f$), and log book-to-market ratio (θ). The parameters in the table are estimated based on the following system:

$$z_t = A + Bz_{t-1} + \varepsilon_t \tag{1.5}$$

For each regression, the estimates and standard errors (in parenthesis) are reported, along with the R^2 and adjusted R^2. Also, for each VAR model, the p-value from the serial correlation test (Breusch-Godfrey LM test) is reported. The Null hypothesis for the serial correlation test is: *no serial correlation exists.* Sample from 1950 - 2019, CRSP and COMPUSTAT.

returns on equity and expected dividend yield.

1.4.2 State-Space Model (Market Returns and Dividend Growth)

Following Campbell and Shiller (1988) [16], by using the log-linearization on price, dividend, and returns, this section presents a state-space model for log excess returns and dividend growth. Increasing the orders for the latent process in the state-space model may not significantly improve the performance since it only adds more correlation terms among the lagged variables, reinforcing the relations among those latent variables (Cochrane, 2008 [23]). Thus, an AR(1) latent process is sufficient for forecasting stock market returns.

Log Linearization

The model starts with the return function, where R_{t+1} denotes the simple return at $t+1$, P_{t+1} denotes the price of the aggregate stock market at $t+1$, and D_{t+1} denotes the dividend at $t+1$.

$$R_{t+1} = \frac{P_{t+1} + D_{t+1}}{P_t} \tag{1.6}$$

Following the first-order Taylor Series, the continuous compounded return can be represented by:

$$\begin{aligned} r_{t+1} &= \ln\left(1 + e^{-dp_{t+1}}\right) + \Delta d_{t+1} + dp_t \\ &\approx \ln\left(1 + \bar{P}/\bar{D}\right) + \frac{-\bar{P}/\bar{D}}{1 + \bar{P}/\bar{D}}\left(dp_{t+1} - \bar{dp}\right) + \Delta d_{t+1} + dp_t \\ &= \kappa - \rho dp_{t+1} + \Delta d_{t+1} + dp_t \end{aligned} \tag{1.7}$$

where $dp_{t+1} \equiv \ln\left(P_{t+1}/D_{t+1}\right)$, $\Delta d_{t+1} \equiv \ln\left(D_{t+1}/D_t\right)$, $\rho \equiv \frac{\bar{P}/\bar{D}}{1+\bar{P}/\bar{D}}$ and $\kappa \equiv \ln(1 + \frac{\bar{P}}{\bar{D}}) + \rho \bar{dp}$. $\bar{D}$ and $\bar{P}$ are the sample average of aggregate dividends and prices, and $\bar{dp}$ is the sample average of linearized dividend-price ratio. Then, the log-linearized dp_t can be written as

$$dp_t = -\frac{\kappa}{1-\rho} + \sum_{j=1}^{\infty} \rho^{j-1} r_{t+j} - \sum_{j=1}^{\infty} \rho^{j-1} \Delta d_{t+j} \tag{1.8}$$

Assumptions for Latent Processes

Similar to Van Binsbergen and Koijen (2008) [88], and Cochrane (2008) [23], I assume that expected returns and dividend growth rates follow the AR(1) process below, where $\mu_t \equiv E(r_{t+1}|I_t)$ denotes the conditional expectations of log return at time $t+1$, $g_t \equiv E(\Delta d_{t+1}|I_t)$ denotes the conditional expectations of log dividend growth rate at time $t+1$, $\varepsilon_{\mu,t+1}$ represents the shocks for expected return at time $t+1$, $\varepsilon_{g,t+1}$ represents the shocks for expected dividend growth rate at time $t+1$, and $\varepsilon_{d,t+1}$ represents the residual for dividend growth rate at time $t+1$.

$$\begin{aligned} \mu_{t+1} &= \phi_\mu \mu_t + \varepsilon_{\mu,t+1} \\ g_{t+1} &= \phi_g g_t + \varepsilon_{g,t+1} \end{aligned} \tag{1.9}$$

I use E_t to denote expectations conditional on the information set (I_t) up to time period t. By taking the conditional expectation, Equation (1.8) becomes:

$$E_t dp_t = -\frac{\kappa}{1-\rho} + E_t \sum_{j=1}^{\infty} \rho^{j-1} \left(r_{t+j} - \Delta d_{t+j} \right) \tag{1.10}$$

$$\Longrightarrow dp_t = \frac{\mu_t}{1-\phi_\mu \rho} - \frac{g_t}{1-\phi_g \rho} - \frac{\kappa}{1-\rho} \tag{1.11}$$

The log dividend growth follows the equation below:

$$\Delta d_{t+1} = g_t + \varepsilon_{d,t+1} \tag{1.12}$$

Then, by substituting the equations back to Equation (1.7), the return can be written as:

$$r_{t+1} = \mu_t + \varepsilon_{d,t+1} - \rho(k_\mu \varepsilon_{\mu,t+1} - k_g \varepsilon_{g,t+1}) \tag{1.13}$$

where $k_\mu \equiv \frac{1}{1-\rho\phi_\mu}$ and $k_g \equiv \frac{1}{1-\rho\phi_g}$.

Measurement & Transition Equations

Given the processes above, we have the following state-space model, where the observable variables are $y_{t+1} = (r_{t+1}, \Delta d_{t+1})'$ and the unobserved variables are $\beta_t = (\mu_t, g_t, \varepsilon_{\mu,t+1}, \varepsilon_{g,t+1}, \varepsilon_{d,t+1})'$.
Measurement Equation:

$$\underbrace{\begin{bmatrix} r_{t+1} \\ \Delta d_{t+1} \end{bmatrix}}_{\equiv y_{t+1}} = \underbrace{\begin{bmatrix} 1 & 0 & -\rho k_\mu & \rho k_g & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}}_{\equiv H} \underbrace{\begin{bmatrix} \mu_t \\ g_t \\ \varepsilon_{\mu,t+1} \\ \varepsilon_{g,t+1} \\ \varepsilon_{d,t+1} \end{bmatrix}}_{\equiv \beta_t} \tag{1.14}$$

Transition Equation:

$$\underbrace{\begin{bmatrix} \mu_t \\ g_t \\ \varepsilon_{\mu,t+1} \\ \varepsilon_{g,t+1} \\ \varepsilon_{d,t+1} \end{bmatrix}}_{\equiv \beta_t} = \underbrace{\begin{bmatrix} \phi_\mu & 0 & 1 & 0 & 0 \\ 0 & \phi_g & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\equiv F} \underbrace{\begin{bmatrix} \mu_{t-1} \\ g_{t-1} \\ \varepsilon_{\mu,t} \\ \varepsilon_{g,t} \\ \varepsilon_{d,t} \end{bmatrix}}_{\equiv \beta_{t-1}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{\mu,t+1} \\ \varepsilon_{g,t+1} \\ \varepsilon_{d,t+1} \end{bmatrix}}_{\equiv v_t} \tag{1.15}$$

For the error terms, the variances are defined as:

$$Q = var\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{\mu,t+1} \\ \varepsilon_{g,t+1} \\ \varepsilon_{d,t+1} \end{bmatrix}\right) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\mu^2 & \sigma_{\mu g} & \sigma_{\mu d} \\ \sigma_{\mu g} & \sigma_g^2 & \sigma_{gd} \\ \sigma_{\mu d} & \sigma_{gd} & \sigma_d^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.16}$$

To simplify the estimating process, covariances are decomposed as:

$$\sigma_{ij} = \sigma_i \times \sigma_j \times \rho_{ij} \qquad \forall i, j \in \{\mu, g, d\} \text{ and } i \neq j \tag{1.17}$$

Estimations for Stock-variable State-Space Model

Given the measurement and transition equations above, we have the following state-space model.

$$\text{Measurement Equation: } y_{t+1} = H\beta_t \tag{1.18}$$

$$\text{Transition Equation: } \beta_t = F\beta_{t-1} + v_t \qquad v_t \overset{i.i.d}{\sim} N(0, Q) \tag{1.19}$$

Following the above models, I utilize the Kalman filter (Hamilton, 1994 [42]) and conditional maximum likelihood estimation to estimate the vector of parameters[1]:

$$(\phi_\mu, \phi_g, \sigma_\mu, \sigma_g, \sigma_d, \rho_{\mu g}, \rho_{\mu d}, \rho_{gd}) \tag{1.20}$$

The optimization problem is solved by using the R package *astsa* [86]. Also, when we consider the reduced form of the state-space model, there are more parameters than can be identified with the observed data. Following Cochrane(2008) [23], Van Binsbergen and Koijen (2010)

[1]See Appendix A.1 for the details of the Kalman Filter

[88], and Rytchkov (2012) [79], I restrict the correlation between expected dividend growth shocks and realized dividend growth shocks to be 0, $\rho_{gd} = 0$.

To examine the performance of state-space models, following Van Binsbergen and Koijen (2010) [88], I compute the R^2 values for returns as:

$$R^2_{\text{Ret}} = 1 - \frac{v\hat{a}r\left(r_{t+1} - \mu_t^F\right)}{v\hat{a}r\left(r_t\right)} \tag{1.21}$$

where $v\hat{a}r$ denotes the sample variance, and μ_t^F is the predictions of expected returns (μ_t).

Table 1.5 shows the conditional maximum likelihood estimates of the parameters for the previous model (Equations (1.14) to (1.20)), including log returns and log dividend yields. Also, this table reports the R^2 (Equation (1.21)) for in-sample one-step-ahead predictions of market returns. The one-step-forward in-sample predictions of market returns are further estimated in Table 1.7 and plotted in Figure 1.1. The out-of-sample predictions are generated in Table 1.9 and Figure 1.2.

From Table 1.5, I find that the conditional expected return is highly persistent over time, with a significant coefficient ($\phi_\mu = 0.9621$) at 5% level. The coefficient of conditional expected dividend growth (ϕ_g) is not significant at 5% level, with a value equal to 0.1788. The R^2 for in-sample predictions of market returns is 12.60%, and I find an R^2 value of 14.04% for dividend growth rates.

Shocks for expected returns and dividend growth rates are smaller than the shock for expected dividend growth rates. It shows that most of the information existing in expected market returns is carried over time, but the expected dividend policy is not significantly predicted due to high volatility.

These results are consistent with Cochrane (2007) [22], Pastor and Stambaugh (2009)[69], and Binsbergen and Koijen (2010) [88], showing that expected returns are more persistent than conditional expected dividend growth. But, the R^2 estimated base on Equation (1.21) is slightly higher than the one from Binsbergen and Koijen (2010) [88], which may be caused by different observable variables and the length of data.

Table 1.5: SSM Estimation Results based on Returns & Dividend Yield

	Estimates	S.E.
ϕ_μ	0.9621	0.0207
ϕ_g	0.1788	0.1235
σ_μ	0.0308	0.0120
σ_g	0.0824	0.1357
σ_d	0.0089	0.0073
$\rho_{\mu g}$	0.8181	0.1091
$\rho_{\mu d}$	-0.0604	1.1004
Market Return R^2	0.1260	
Dividend Growth R^2	0.1404	

This table reports the estimations of parameters for the state-space model based on the log returns (μ_t) and log dividend growth rate (g_t) from Equations (1.14) to (1.20). The restriction, $\rho_{gd} = 0$, is implemented. The models are estimated by conditional maximum likelihood using sample data from 1950 - 2019, CRSP and COMPUSTAT. This table also reports the R^2 (Equation (1.21)) for one-step-ahead predictions (in-sample) of market returns and dividend growth rate. The one-step-forward predictions of market returns are plotted in Figure 1.1.

1.4.3 State-Space Model (Market Returns and Returns on Equity)

Following Campbell and Shiller (1998a) [16], Vuolteenaho (2000) [90], and Vuolteenaho (2002) [91], by assuming the clean-surplus identity, this section presents a State-Space-Model based on returns, returns on equity, and book-to-market ratios.

Clean Surplus Identity

Vuolteenaho (2000) [90] constructs the clean-surplus identity based on book equity (B_t), earnings (X_t) and dividends (D_t).

$$B_t = B_{t-1} + X_t - D_t \tag{1.22}$$

Based on the clean surplus accounting identity, Equation (1.22), and return on equity (ROE):

$$E_t = X_t/B_{t-1} \tag{1.23}$$

we can define log returns for market and accounting as:

$$r_t + f_t \equiv \ln\left(\frac{M_t + D_t}{M_{t-1}}\right) = \ln\left(1 + \frac{\Delta M_t + D_t}{M_{t-1}}\right) = \ln\left(1 + R_t + F_t\right) \tag{1.24}$$

$$e_t \equiv \ln\left(\frac{B_t + D_t}{B_{t-1}}\right) = \ln\left(1 + \frac{\Delta B_t + D_t}{B_{t-1}}\right) = \ln\left(1 + E_t\right) \tag{1.25}$$

where M_t denotes the market value at time t, F_t denotes the interest rate at time t, f_t denotes $\ln(1+F_t)$ at time t, and D_t denotes the dividend at time t.

Log Linearization

By defining $\delta_t = d_t - b_t$, where d_t is the log dividend at time t and b_t is the log book equity at time t, the log market returns can be written as:

$$\begin{aligned} r_t + f_t &= \ln\left(\exp\left(\ln\frac{M_t}{D_t}\right) + 1\right) + \ln\left(D_t/D_{t-1}\right) + \ln\left(D_{t-1}/M_{t-1}\right) \\ &= \ln\left(\exp\left(-\delta_t\right) + 1\right) + \Delta d_t + \delta_{t-1} \end{aligned} \tag{1.26}$$

By defining $\gamma_t = d_t - m_t$, where m_t is the log market value at time t, the log accounting returns can be written as:

$$\begin{aligned} e_t &= \ln\left(\exp\left(\ln\frac{B_t}{D_t}\right) + 1\right) + \ln\left(D_t/D_{t-1}\right) + \ln\left(D_{t-1}/B_{t-1}\right) \\ &= \log\left(\exp\left(-\gamma_t\right) + 1\right) + \Delta d_t + \gamma_{t-1} \end{aligned} \tag{1.27}$$

Then, by using Equation (1.27) to subtract Equation (1.26), the identity can be approximated based on the first-order Taylor Series as:

$$\begin{aligned} e_t - r_t - f_t &= \ln(1+\bar{B}/\bar{D}) + \frac{-\bar{B}/\bar{D}}{1+\bar{B}/\bar{D}}(\gamma_t - \bar{\gamma}) - \ln(1+\bar{M}/\bar{D}) - \frac{-\bar{M}/\bar{D}}{1+\bar{M}/\bar{D}}\left(\delta_t - \bar{\delta}\right) - \theta_{t-1} \\ &\approx \alpha + \rho\theta_t - \theta_{t-1} + \kappa_t \end{aligned} \tag{1.28}$$

where θ_t denotes the log book to market ratio, α denotes a constant parameter, ρ denotes the discount ratio (a constant parameter smaller than 1), and κ_t denotes the approximation errors [2].

Similar to the log linearization for dividend yield, the log-linearized book-to-market ratio can be written as:

$$\theta_{t-1} = \frac{\alpha}{1-p} + \sum_{j=0}^{\infty} \rho^j r_{t+j} - \sum_{j=0}^{\infty} \rho^j e^*_{t+j} \tag{1.29}$$

where $e^*_{t+j} \equiv e_t - \kappa_t - f_t$, which denotes the log excess return on equity.

Assumptions for Latent Processes

Similar to the state-space model above, I assume that expected returns and excess return on equity follow the AR(1) process below, where $\mu_t \equiv E(r_{t+1}|I_t)$, $h_t \equiv E(e^*_{t+1}|I_t)$, $\varepsilon_{\mu,t+1}$ represents the shocks for expected return at time $t+1$, $\varepsilon_{h,t+1}$ represents the shocks for expected excess return on equity at time $t+1$, $\varepsilon_{e,t+1}$ represents the approximation errors for excess return on equity at time $t+1$.

$$\begin{aligned} \mu_{t+1} &= \phi_\mu \mu_t + \varepsilon_{\mu,t+1} \\ h_{t+1} &= \phi_h h_t + \varepsilon_{h,t+1} \end{aligned} \tag{1.30}$$

[2] See Appendix A.2 for approximating κ_t in detail.

By taking the conditional expectation based on the information set, I_t, the log-linearized book-to-market ratio is:

$$\theta_t = \frac{\alpha}{1-\rho} + \frac{\mu_t}{1-\phi_\mu \rho} - \frac{h_t}{1-\phi_h \rho} \tag{1.31}$$

The log excess return on equity follows the equation below:

$$e^*_{t+1} = h_t + \varepsilon_{e,t+1} \tag{1.32}$$

After plugging the result back into Equation (1.28), the identity becomes:

$$r_{t+1} = \mu_t + \varepsilon_{e,t+1} + \rho\left(k_h \varepsilon_{h,t+1} - k_\mu \varepsilon_{\mu,t+1}\right) \tag{1.33}$$

where $k_h \equiv \frac{1}{1-\rho\phi_h}$ and $k_\mu \equiv \frac{1}{1-\rho\phi_\mu}$. Then, we no longer have the dividends existing in the model.

Measurement & Transition Equations

Given the processes above, we have the following state-space model, where the observable variables are $y_{t+1} = (r_{t+1}, e^*_{t+1})'$ and the unobserved variables are $\beta_t = (\mu_t, h_t, \varepsilon_{\mu,t+1}, \varepsilon_{h,t+1}, \varepsilon_{e,t+1})'$.

Measurement Equation:

$$\underbrace{\begin{bmatrix} r_{t+1} \\ e^*_{t+1} \end{bmatrix}}_{\equiv y_{t+1}} = \underbrace{\begin{bmatrix} 1 & 0 & -\rho k_\mu & \rho k_h & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}}_{\equiv H} \underbrace{\begin{bmatrix} \mu_t \\ h_t \\ \varepsilon_{\mu,t+1} \\ \varepsilon_{h,t+1} \\ \varepsilon_{e^*,t+1} \end{bmatrix}}_{\equiv \beta_t} \tag{1.34}$$

Transition Equation:

$$\underbrace{\begin{bmatrix} \mu_t \\ h_t \\ \varepsilon_{\mu,t+1} \\ \varepsilon_{h,t+1} \\ \varepsilon_{e^*,t+1} \end{bmatrix}}_{\equiv \beta_t} = \underbrace{\begin{bmatrix} \phi_\mu & 0 & 1 & 0 & 0 \\ 0 & \phi_h & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\equiv F} \underbrace{\begin{bmatrix} \mu_{t-1} \\ h_{t-1} \\ \varepsilon_{\mu,t} \\ \varepsilon_{h,t} \\ \varepsilon_{e^*,t} \end{bmatrix}}_{\equiv \beta_{t-1}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{\mu,t+1} \\ \varepsilon_{h,t+1} \\ \varepsilon_{e^*,t+1} \end{bmatrix}}_{\equiv v_t} \tag{1.35}$$

For the error terms, the variances are defined as:

$$Q = var\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{\mu,t+1} \\ \varepsilon_{h,t+1} \\ \varepsilon_{e^*,t+1} \end{bmatrix} \right) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\mu^2 & \sigma_{\mu h} & \sigma_{\mu e^*} \\ \sigma_{\mu h} & \sigma_h^2 & \sigma_{he^*} \\ \sigma_{\mu e^*} & \sigma_{he^*} & \sigma_{e^*}^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.36}$$

To simplify the estimating process, similar to the previous state-space model, covariances are decomposed as:

$$\sigma_{ij} = \sigma_i \times \sigma_j \times \rho_{ij} \qquad \forall i, j \in \{\mu, h, e^*\} \text{ and } i \neq j \tag{1.37}$$

Estimations for Accounting-variable State-Space Model

Given the measurement and transition equations above, we have the following state-space model.

$$\text{Measurement Equation: } y_{t+1} = H\beta_t \tag{1.38}$$

$$\text{Transition Equation: } \beta_t = F\beta_{t-1} + v_t \qquad v_t \overset{i.i.d}{\sim} N(0, Q) \tag{1.39}$$

To simplify the estimating process, I utilize the Kalman filter (Hamilton, 1994 [42]) and conditional maximum likelihood estimation to estimate the vector of parameters:

$$(\phi_\mu, \phi_h, \sigma_\mu, \sigma_h, \sigma_{e^*}, \rho_{\mu h}, \rho_{\mu e^*}, \rho_{he^*}) \tag{1.40}$$

The optimization problem is solved by using the R package *astsa* [86]. Similar to the previous state-space model, I restrict the correlation between expected ROE shocks and realized ROE shocks to be 0, $\rho_{he} = 0$.

Table 1.6 shows the conditional maximum likelihood estimates of the parameters for the previous model (Equations (1.34) to (1.40)), including log returns and log returns on equity. Also, this table reports the in-sample one-step-forward predictions R^2 for market returns and dividend growth. The one-step-forward in-sample predictions are further estimated in Table 1.7 and plotted in Figure 1.1. The out-of-sample predictions are generated in Table 1.10 and Figure 1.2.

From Table 1.6, I find that the conditional expected return and returns on equity are highly persistent over time, with significant coefficients ($\phi_\mu = 0.8886$ & $\phi_h = 0.9619$) at 5% level. Different from the expected dividend growth rate, the expected return on equity is more persistent than expected market returns, which supports that aggregate earnings are more stable than dividend policies. Intuitively, if firms have a large log return on equity in the last period, firms will have more returns on equity for the next period.

Also, the correlation between expected returns and returns on equity is 0.8647, which is larger and more significant than the correlation between expected returns and dividend growth rates, 0.8181. It shows while considering firms as a whole, firms with higher expected return on equity will lead to higher expected returns, which provide more useful information when forecasting the market returns.

In Table 1.6, the R^2 for in-sample predictions of market returns is 17.85%, and I find an R^2 value of 23.63% for returns on equity. Comparing the results with Table 1.5, I find that this R^2 for one-step-forward predictions of market returns is larger. The dividend yield or dividend-price ratio contains the information of future stock market returns, but with unstable dividend policies, the predictions perform poorer than the accounting variables. Also, the R^2 for returns on equity is larger than the R^2 for dividend growth rates, which further shows that accounting data are more persistent than dividends. Section 1.5 will further analyze the results.

Table 1.6: SSM Estimation Results based on Returns & Returns on Equity

	Estimates	S.E.
ϕ_μ	0.8886	0.1043
ϕ_h	0.9619	0.0359
σ_μ	0.0237	0.0187
σ_h	0.0202	0.1049
σ_e	0.0762	0.0070
$\rho_{\mu h}$	0.8647	4.2596
$\rho_{\mu e}$	-0.1598	0.1735
Market Return R^2	0.1785	
Return on Equity R^2	0.2363	

This table reports the estimations of parameters for the state-space model based on the log returns (μ_t) and log returns on equity (e_t^*) from Equations (1.34) to (1.40). The restriction, $\rho_{he} = 0$, is implemented. The models are estimated by conditional maximum likelihood using sample data from 1950 - 2019, CRSP and COMPUSTAT. This table also reports the R^2 (Equation (1.21)) for one-step-ahead predictions (in-sample) of market returns and dividend growths. The one-step-forward predictions of market returns are plotted in Figure 1.1.

1.5 Results

In this section, I compare and discuss the estimated results from previous models. Then, I decompose variances of price-dividend ratio, book-to-market ratio, and unexpected returns in both state-space models. Also, by splitting the sample as training and testing datasets, this part tests the out-of-sample predictions' stability of state-space models.

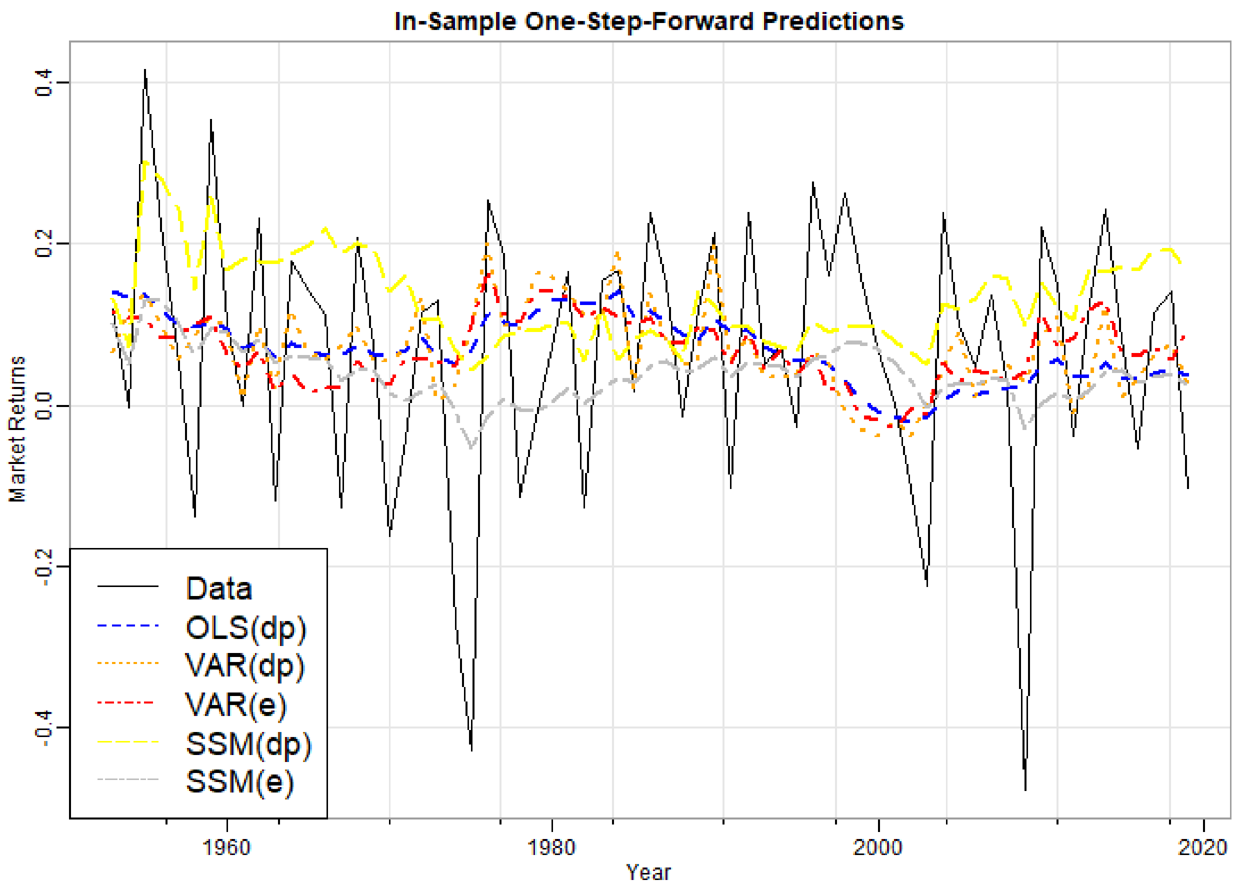
1.5.1 In-sample Predictions

Table 1.7 shows the in-sample one-step-forward prediction R^2 for all the methods mentioned above. The linear regressions of returns on the dividend-price ratio or book-to-market ratio show the poorest R^2, but with significant coefficients. These results are similar to past results, such as Van Binsbergen and Koijen (2010) [88], Goyal and Welch (2008) [37], Cochrane (2007) [22], and Vuolteenaho (2002) [90]. Further, predictions from VAR models show the possibility of depicting the stock market with the order selected based on information criterion. While estimating VAR models, existing unit roots in the dividend-price and book-to-market ratio weaken the prediction results, which is also shown by the serial correlation tests from Table 1.4. The state-space models give better in-sample predictions than previous ones, which can roughly capture the main movements of market returns.

Based on the results from Table 1.5 and 1.6, it is shown that the unexpected variables capture some important parts of the return predictions. The evolving latent process for the state space model with market returns and dividend growth is similar to the results from Van Binsbergen and Koijen (2010), showing more persistence in expected market returns than dividend growth rates. For the state-space model based on market returns and returns on equity, it generates the best in-sample predictions. Also, it shows persistence both in expected market returns and returns on equity, which improves the forecastability of the predictive system.

Figure 1.1 plots all the in-sample predicted values from different models. OLS and VAR share common trends in predictions of market returns. The state-space model with stock variables fluctuates around the initial value of expected market returns, which does not capture the volatility of returns over time. However, the accounting-based state-space model follows the movement of market returns in the 70 years, which reinforces the statement that accounting data can help to improve the performance in predicting market returns.

Figure 1.1: In-sample One-step-forward Predictions



This figure plots the one-step-forward predictions based on different models. OLS(dp) denotes the predictions based on table 1.3, VAR(dp, dp) denotes the predictions based on the short VAR model (including returns, dividend yield, and dividend growth rate), VAR(ROE, b/m ratio) denotes the short VAR model (including returns, returns on equity, and book-to-market ratio), SSM(dp) is the state-space model based on returns and dividend-price ratio, and SSM(e) denotes the state-space model based on returns and return on equity. The models are estimated by conditional maximum likelihood using sample data from 1950 - 2019, CRSP and COMPUSTAT

Table 1.7: In-Sample Predictions R^2

	R^2
OLS(dp)	0.0618
OLS(bm)	0.0572
VAR(dp, dg)	0.1090
VAR(e, theta)	0.0659
SSM(dp)	0.1260
SSM(e)	0.1785

This table reports the R^2 (Equation (1.21)) for one-step-ahead predictions based on different models. OLS(dp) and OLS(bm) denotes the predictions based on the first and fourth regressions in table 1.3, VAR(dp, dp) denotes the predictions based on the short VAR model (including returns, dividend yield, and dividend growth rate), VAR(e, theta) denotes the short VAR model (including returns, returns on equity, and book-to-market ratio), SSM(dp) is the state-space model based on returns and dividend-price ratio, and SSM(e) denotes the state-space model based on returns and return on equity. The predictions are plotted in Figure 1.1. The models are estimated by conditional maximum likelihood using sample data from 1950 - 2019, CRSP and COMPUSTAT.

1.5.2 Variance Decomposition

Following Campbell and Shiller (1988a, 1988b) [16] [17], Vuolteenaho (2002) [91] and Binsbergen and Koijen (2010) [88], I derive variance decompositions of price-dividend ratio, book-to-market ratio, and unexpected returns in both state-space models. The variance decomposition for the price-dividend ratio is given as:

$$\begin{aligned} \operatorname{var}(dp_t) = &\left(\frac{1}{1-\phi_\mu\rho}\right)^2 \operatorname{var}(\mu_t) + \left(\frac{1}{1-\phi_g\rho}\right)^2 \operatorname{var}(g_t) \\ &- \frac{2}{(1-\phi_\mu\rho)(1-\phi_g\rho)} \operatorname{cov}(\mu_t, g_t) \end{aligned} \tag{1.41}$$

The first term denotes the variation due to expected returns (discount rate news); the second term measures the variation due to expected dividend growth rates (cash-flow news); and the last term represents the covariation between these variation.

For the book-to-market ratio, the variance decomposition is given as:

$$\begin{aligned}\operatorname{var}(\theta_t) = &\left(\frac{1}{1-\phi_\mu\rho}\right)^2 \operatorname{var}(\mu_t) + \left(\frac{1}{1-\phi_h\rho}\right)^2 \operatorname{var}(h_t) \\ &- \frac{2}{(1-\phi_\mu\rho)(1-\phi_h\rho)} \operatorname{cov}(\mu_t, h_t)\end{aligned} \tag{1.42}$$

Similar to the price-dividend ratio, the book-to-market ratio is decomposed into three terms: variation due to expected returns (discount rate news), variation due to expected returns on equity (cash-flow news), and the covariation between these two news.

In Panel A from Table 1.8, I show the results of variance decompositions for price-dividend and book-to-market ratios. Following Binsbergen and Koijen (2010) [88], I standardize the right-hand side of Equation (1.41) & (1.42), and the sum of three terms is 100%. Similar to the results from Binsbergen and Koijen (2010) [88], I find that most of the variation in the price-dividend ratio is related to expected return variation. However, I find that both expected market returns and expected returns on equity play important roles in affecting variation of book-to-market ratios.

Further, I decompose the variation of unexpected aggregate stock returns with dividend growth rate news and market returns news as:

$$\begin{aligned}\operatorname{var}(r_{t+1}-\mu_t) = &(\rho k_\mu)^2 \operatorname{var}(\varepsilon_{\mu,t+1}) + \operatorname{var}(\varepsilon_{d,t+1} + \rho k_g \varepsilon_{g,t+1}) \\ &- 2\rho k_\mu \operatorname{cov}(\varepsilon_{\mu,t+1}, \varepsilon_{d,t+1} + \rho k_g \varepsilon_{g,t+1})\end{aligned} \tag{1.43}$$

This equation follows similar algorithms as above, where the variance of unexpected returns is decomposed into three parts: variation in discount rate news, variation in cash-flow news, and co covariation between these two components. The second term groups the news from real and expected dividend growth rates together.

For the decomposition of variance of unexpected aggregate stock returns with returns on equity and market returns, it is given as:

$$\begin{aligned} \mathrm{var}\left(r_{t+1}-\mu_t\right) = & \left(\rho k_\mu\right)^2 \mathrm{var}\left(\varepsilon_{\mu,t+1}\right) + \mathrm{var}\left(\varepsilon_{e,t+1}+\rho k_h \varepsilon_{h,t+1}\right) \\ & - 2\rho k_\mu \,\mathrm{cov}\left(\varepsilon_{\mu,t+1}, \varepsilon_{e,t+1}+\rho k_h \varepsilon_{h,t+1}\right) \end{aligned} \tag{1.44}$$

As before, I group the news from real and expected returns on equity together to form the cash-flow news (second term). Then, I compute the influence of discount news, cash-flow news, and the covariance between these two terms.

In Panel B from Table 1.8, I show the results of variance decompositions for unexpected returns with two different variable sets, dividend growth rates and returns on equity. Similar to Panel A, I standardize all terms on the right-hand size of Equations (1.43) & (1.44), so the sum of these terms is 100%.

In Panel B, for the variance decomposition with returns on equity, cash-flow shocks play a more significant role in affecting the variance of unexpected market returns. Also, the correlation between discount rate and cash-flow news is higher than the one with dividend growth rates. The difference comes from the persistence of expected returns on equity (ϕ_h), which is larger and more significant than the expected dividend growth rates as shown in Table 1.5 & 1.6. Finally, using accounting data not only improves the forecastability of market returns but also suggests that cash-flow news plays a profound role in explaining the variation of unexpected market returns.

1.5.3 Out-of-sample Predictions

To generate out-of-sample predictions, I split the sample data into training and testing datasets. The training dataset contains the data from 1950 to 2012, and the testing dataset contains the data from 2013 to 2019.

Table 1.9 shows the parameters estimated based on the state-space model (market returns and dividend growth rates) with training(in-sample) and testing(out-of-sample) dataset. It has similar results as the one with the entire dataset (Table 1.5) in Section 1.4.2. Expected

Table 1.8: Variance Decompositions of the Price-Dividend/Book-to-Market Ratio and Unexpected Market Returns

	Discount Rates	Cash Flows	Covariance
Panel A: Decomposition of Price-Dividend/Book-to-Market Ratio			
Price-Dividend Ratio	115.6%	2.47%	-18.2%
Book-to-Market Ratio	76.5%	154.2%	-130.6%
Panel B: Decomposition of Unexpected Market Returns			
Returns & Dividends	135.8%	2.8%	-38.6%
Returns & Returns on Equity	9.5%	147.9%	-57.43%

This table reports variance decompositions of price-dividend ratio, book-to-market ratio, and unexpected returns. "Discount Rates" refers to variation due to expected return variation, "Cash Flows" refers to the variation due to expected dividend growth rates or expected returns on equity variation, and "covariance" refers to the covariation between these two terms. In Panel A, I present variance decompositions of price-dividend ratio and book-to-market ratio based on Equation (1.41) & (1.42). In Panel B, I present variance decompositions of unexpected returns with different variables based on Equation (1.43) & (1.44). The models are estimated by conditional maximum likelihood using sample data from 1950 - 2019, CRSP and COMPUSTAT.

returns are still more persistent than expected dividend growth rate over time. Also, as it has a smaller dataset, R^2 for in-sample predictions is smaller than the one in Table 1.5.

However, I find a high R^2 for out-of-sample predictions, 31.42%. When predicting the market returns, the state-space model perform well for short-range forecasting. Because expected returns are persistent over time, with a significantly estimated coefficient (ϕ_μ), most of the out-of-sample market returns are predicted by the model.

Table 1.9: Out-of-Sample SSM Estimated Results based on Dividend Yield & Returns

	Estimates	S.E.
ϕ_μ	0.9574	0.0230
ϕ_g	0.1896	0.3089
σ_μ	0.0313	0.0146
σ_g	0.0443	0.4993
σ_d	0.1081	0.2045
$\rho_{\mu g}$	0.8749	3.613
$\rho_{\mu d}$	0.07518	2.3398
In-sample Market Returns R^2	0.0611	
Out-of-sample Market Returns R^2	0.3142	

This table reports the estimations of parameters for the state-space model based on the log returns and log dividend growth rate from equations (1.14) to (1.20). The restriction, $\rho_{gd} = 0$, is implemented. The models are estimated by conditional maximum likelihood using sample data from 1950 - 2012 and tested on sample data from 2013 - 2019, CRSP and COMPUSTAT. This table also reports the R^2 (Equation (1.21)) for the one-step-forward predictions (in-sample) and one-step-forward predictions. The one-step-forward predictions are plotted in Figure 1.2.

Table 1.10 shows the parameter and R^2 estimated based on the state-space model (market returns and return on equity) with training(in-sample) and testing(out-of-sample) dataset. The results are similar to the previous results from Table 1.6 based on the full sample. Also, expected returns on equity are still more persistent than expected returns over time. Similarly to the stock-variable based state-space model, with a smaller dataset, it shows a smaller R^2 for in-sample predictions.

Also, I find the R^2, 35.62%, for out-of-sample predictions is larger than the one from the state-space model with returns and dividends. It shows that, with significant and persistent state variables (ϕ_μ, ϕ_h), more information of market returns is captured by the latent predictive process in the accounting-based state-space model. This is also part of the evidence that accounting earnings are more stable than dividend growth in market returns.

Table 1.10: Out-of-Sample SSM Estimated Results based on Returns on Equity & Returns

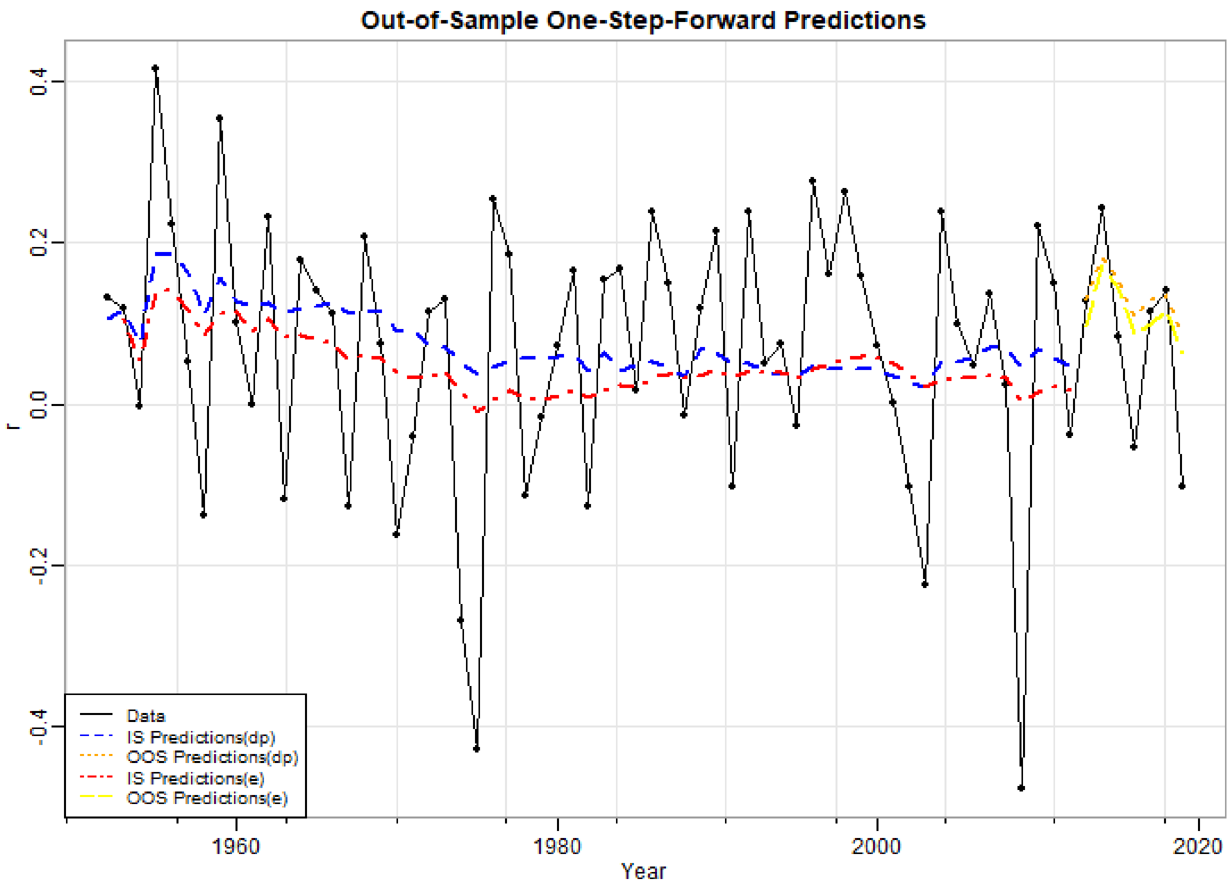
	Estimates	S.E.
ϕ_μ	0.9353	0.1177
ϕ_h	0.9815	0.0365
σ_μ	0.0141	0.296
σ_h	0.0131	0.0085
σ_e	0.0688	0.0066
$\rho_{\mu h}$	0.8108	3.4930
$\rho_{\mu e}$	-0.3318	0.5706
In-sample Market Returns R^2	0.0920	
Out-of-sample Market Returns R^2	0.3562	

This table reports the estimations of parameters for the state-space model based on the log returns and log dividend growth rate from equations (1.34) to (1.40). The restriction, $\rho_{he} = 0$, is implemented. The models are estimated by conditional maximum likelihood using sample data from 1950 - 2012 and tested on sample data from 2013 - 2019, CRSP and COMPUSTAT. This table also reports the R^2 (Equation (1.21)) for the one-step-forward predictions (in-sample) and one-step-forward predictions. The one-step-forward predictions are plotted in Figure 1.2.

Figure 1.2 further supports the statements above. The in-sample predictions for the state-space model with returns and dividends varies around the mean of market returns, while the in-sample predictions for the state-space model with market returns and returns on equity captures most of the fluctuations over time. The out-of-sample predictions are similar for the two state-space models, but the accounting-based model captures more information in the market returns by showing similar movements over time.

As a result, this section shows that the state-space model performs well and is stable

Figure 1.2: Out-of-sample One-step-forward Predictions



This plot shows the in-sample and out-of-sample one-step-forward predictions based on dividend-price ratio or log returns on equity. IS Predictions (dp) & OOS Predictions (dp) are estimated based on the state-space model of returns and dividend-price ratio. IS Predictions (e) & OOS Predictions (e) are estimated based on the state-space model of returns and returns on equity. The models are estimated by conditional maximum likelihood using sample data from 1950 - 2012 and tested on sample data from 2013 - 2019, CRSP and COMPUSTAT.

on the sample. Accounting variables have better performance than dividend yields due to the unstable dividend policies over time. Though the out-of-sample predictions are similar, the forecastability of expected returns on equity can improve the performance of accounting-based state-space model in the long-range predictions. Also, when there is a truncated dataset, the estimated coefficients may vary based on different time ranges, but the performance seems to be consistent.

1.6 Conclusion

In this paper, I test the performance of different models based on the identities of stock market variables and accounting variables. I also estimate a present-approach to predict annual market returns by assuming several latent variables, including expected market returns, and expected dividend growth rates. The model is combined with the log-linearized identity to drive the dynamics of variables behind the observations and utilize the Kalman filter to estimate the movements. Comparing with VAR models, state-space models handle the instability of data and dividend policy by forming evolving latent processes, which reduce the uncovered information in residuals.

Then, I propose a new approach with accounting variables based on the clean surplus identity from past literature. Accounting variables are more stable than the stock market variables while considering the changes and differences in dividend policy. As a result, the latent processes based on accounting variables further improve the forecastability of future market returns. Also, cash-flows news based on expected returns on equity is more persistent than expected market returns. Estimated results from in-sample and out-of-sample predictions show that new approaches can improve the forecastability of future returns.

These approaches generate latent processes for expected market returns and involve predictions in the measurement equation by introducing the log linearization of market returns. In this case, the information existing in the uncovered state variables is used while updating the predicting system. For the latent dynamics, it could be useful to incorporate more variables, but it would be much more complex to build the correct identity among the variables. For long run, predictions gradually approach to averages of sample data due to the mitigation of state variables. And, the filtering system can be further extended with other algorithms, considering the improvements in computational tools. Incorporating more advanced techniques, such as machine learning or deep learning, can increase the accuracy of selecting features or updating the existing system, especially the long-short-term memory networks. Also, the increasing availability of data could help to build different predicting systems with different linear constraints.

Chapter 2

PROBABILISTIC DEEP-LEARNING MODELS FOR PREDICTING U.S. MARKET RETURNS

2.1 Introduction

Forecasting the aggregate market equity returns is one of the main questions in modern empirical asset pricing research. Researchers have been creating different information sets to use, including variables such as sentiments, news, and various indicators. Given a large number of variable sets, traditional models may not work well, as there may be overfitting or structural problems. Recently, machine learning has been introduced into economic literature, such as Gu et al. (2020) [40]. Moreover, considering the evolving processes of time series, researchers study multiple dependencies across both macro and individual effects (Salinas et al. (2020) [83], Rangapuram et al. (2018) [71], Salinas et al. (2019) [82], Wang et al. (2019) [92]). For example, the U.S. stock market returns depend both on U.S. economy and firms' financial statements.

In this paper, I introduce advanced models based on recurrent neural networks (RNN) with long short-term memory (LSTM). Also, I conduct a comparative analysis of these methods for forecasting U.S. market returns. My primary contributions are two-fold. First, this is the first paper that introduces new state-of-art models, probabilistic deep learning models, into asset pricing, which significantly improves the predictability of market returns. The high out-of-sample R^2 ranges from 42.33% to 65.80% with different information sets.

Probabilistic deep learning models can be regarded as extensions of classical methods, such as autoregressive (AR), vector autoregressive (VAR), or state-space models (SSM), which build the link between traditional metrics and advanced neural nets. Classical models must be re-trained or re-fitted for each set of time series, which is complex while forecasting multiple time series together[1]. This may fail to optimize predicting accuracy, especially when people consider the covariation among multiple time series and temporal dynamics with seasonality.

However, deep-learning models discover more information by automatically building structures and sharing covariation across different time series. These processes reduce labor efforts of manually computing sophisticated structures. For example, the deep state-space model (DSSM) automatically learns the latent dynamic systems by using the outputs from RNNs, which reduces potential bias or inaccurate knowledge of internal latent states. This also helps to reduce the requirement of having long historical data. Incorporating deep learning into classical models also alleviates strict assumptions, such as homoskedasticity or stationarity, especially for multivariate time series forecasting. Further, by estimating the conditional distribution of future market returns, probabilistic deep learning models provide more interpretability and flexibility than other neural nets.

Second, I analyze and compare various methods within the family of probabilistic deep-learning models and with results from classical time series models. Then, I use the same dataset to design experiments, including both stock and accounting variables. Similarly to the analysis conducted in the last chapter, I find that accounting variables are more informative than stock variables when using advanced deep learning models. Deep AR model performs the best when using aggregate stock variables(market returns, dividend growth rates, and divided-price ratios) with 42.33% R^2. When using accounting variables(market returns, returns on equity, book-to-market ratios), deep state-space model performs the best with 50.92% R^2. Deep AR models with both stock and accounting variable reports the highest R^2, 65.80%.

While considering the capture of future market movements, deep state-space and deep AR models perform well even with a relatively small dataset. But, deep VAR and deep factor models may require larger datasets to improve the accuracy of predicting market

[1]See Gu et al. (2020) [40] for their way of fitting models to different datasets.

returns. Also, I conduct robust analysis based on different train-test split, 90:10 and 85:15 ratios. Increasing the length of datasets generates better performance statistics for most of the models. Nevertheless, simply increasing data size does not help to improve model performance. Probabilistic deep learning models do not perform well with large amount of noise in the data, and feature selection must be applied to generate informative variables. Also, as probabilistic deep learning models automatically learn structure of latent states, their performance may not be significantly affected by the size of data. Chapter three further analyzes these models in detail.

The organization of this paper is as follows. In Section 2.2, I review the background of machine learning and deep learning in the existing literature, including historical forecasting approaches and potential flaws. Section 2.3 discusses four different deep-learning models for forecasting market returns through intuitive, structural, and tuning perspectives. In Section 2.4, I outline the empirical findings based on U.S. stock market returns, including data manipulations, empirical results, and discussions of methods. Section 2.5 contains my conclusions and further steps.

2.2 Literature Review

With the rapid development of computational technologies, researchers have been widely using machine learning models in financial econometrics. Machine learning can help improve asset pricing models' performance, uncover casual relationships, and predict future movements. For example, Rapach et al. (2013) [72] utilize the LASSO to show that lagged U.S. market returns can significantly predict market returns in other countries. Varian (2014) [89], Kelley et al.(2019) [52], and Giglio and Xiu (2021) [35] use machine learning to improve the accuracy of feature selections to handle "the curse of dimensionality" with large economic datasets. Mullainathan and Spiess (2017) [66] mention that introducing different algorithms in machine learning can improve the predictability and estimations in economics. Similarly, Gu et al. (2020) [40] state that machine learning methods profoundly help to measure asset risk premia. By allowing for nonlinearities and vast predictor sets, they compare different approaches to figure out the best model and most potent variables.

Following three subsections briefly summarize the past literature in machine learning, deep learning, and probabilistic deep learning models.

2.2.1 Machine Learning for Asset Pricing

When researchers apply machine learning methods in social science, besides techniques, there are other ethical problems that have to be considered. Bartlett (2020) [7] and Morse and Pence (2021) [65] argue that machine learning and artificial intelligence can discriminate against marginalized groups in different areas, such as housing mortgage and credit determination. Also, the choices of algorithms made by analysts or programmers can still discriminate against the public in financial markets. Though some methods such as sequential learning and causal modeling can help, it is easy to ignore potential bias. However, while applying machine learning to financial economic research, especially asset pricing, researchers focus on estimations and predictions, mostly unrelated to potential ethical problems. And people can utilize the advantages of these advanced technologies.

Compared with traditional econometrics, machine learning has some conflicts with financial economic research. Generally, machine learning focuses more on developing algorithms to make predictions (Wu et al. (2008) [94]), but traditional econometrics is "to specify a target, an estimand, that is a function of a joint distribution of the data" (Athey and Imbens (2019) [3]). Also, traditional econometrics form relationships among variables based on specific assumptions, but machine learning does not have a lot of assumptions about the model or data structure. Then, researchers must validate their results to balance bias and variance due to the flexibility.

As machine learning model can produce excellent forecasts, they have the potential of making significant progress in predicting excess stock returns, especially considering the accumulated predictors or methods in the past century. Researchers, such as Cochrane (2007) [22] and Goyal and Welch (2008) [37], have been arguing the predictability of excess stock returns for years. Though Cochrane (2007) [22] defends the forecastability of excess market returns by showing that dividend growth rates are not predictable, researchers are still attempting to find different ways to depict or estimate stock markets. Considering the wide list of predictors or information sets, machine learning is ideal for selecting features and reducing dimensionality due to its high flexibility (Gu et al. (2020) [40]). Also, different from traditional approaches, the data mining process in machine learning requires relatively larger datasets, which may be a problem in some fields with short history.

When estimating models to predict asset returns, most common methods are built under linear assumptions among variables, such as the vector autoregressive (VAR). Researchers also use penalized linear methods, such as least absolute shrinkage and selection operator (LASSO), to avoid overfitting large datasets, in which computers automatically focus more on the relevant predictors instead of the least relevant ones. Moreover, one of the complex problems is the vague format of functional forms based on different sets of information. Then, besides linear models with penalization, other nonlinear methods are worth to be considered, including tree-based models (boosted trees and random forests), dimension reduction, deep learning, and neural nets.

To apply machine learning to time series data, researchers must figure out the number of lags for each variable. Due to the non-convex optimization problem of linear or penalized linear regressions in time series, tree-based models and neural networks perform the best for predictions in general (Gu et al., 2020, [40]). Chen et al. (2021) [19], Feng et al. (2018) [30], Gu et al. (2021) [41], Heaton et al. (2017) [43], and Benzoni (2011) [8] also apply deep learning in asset pricing to improve the performance of predictions or exploit the relationships across assets or portfolios. These researches further show that deep learning can help to solve the challenges due to large datasets or standard methods, as deep learning can detect hidden interactions inside the market.

2.2.2 Deep Learning for Asset Pricing

Deep learning neural networks, one of the most powerful models in machine learning, has been efficiently used in both academia and industry to solve complex problems, such as natural language process (NLP) and artificial intelligence (Deep Blue and Alpha Go). Neural nets build several nonlinear layers to make predictions through different interactions. In general, the tuning process of neural networks requires substantial efforts.

Economists started studying neural nets in econometric methods around the 1990s (Hornik et al. (1989) [48]; Cybenko (1989) [24]; Athey and Imbens (2019) [3]). However, due to the computational costs, neural networks were not popular in finance until recent years. For asset pricing, deep learning provides a powerful framework due to its capability to extract nonlinear relationships and handling complex structures of large datasets (Goodfellow et al. (2016) [36], Feng et al. (2020) [31]).

One of the essential deep learning models is the feedforward neural network (FNN) (Hornik et al. (1989) [48]), in which information is carried over activation functions, intermediate(hidden) layers, and evaluated at the output layer[2]. For time series or sequential data, recurrent neural networks (RNN) (Rumelhart et al. (1986) [78]) perform the best, as they can recall and re-scale information from its internal states and automatically influence the current input and output based on historical data[3]. By sharing parameters, recurrent neural networks can analyze samples with different frequencies and share information across them.

In recurrent neural networks, the long short-term memory (LSTM) model (Hochreiter and Schmidhuber (1997) [46]) is one of the most effective models, especially for time sequential data. By introducing LSTM-RNNs, researchers can prevent models from getting ill-conditioned results due to vanishing or exploding gradients used by the stochastic gradient descent algorithms used to fit the models. Also, instead of simply applying nonlinearities to the transformed inputs and recurrent units, LSTM brings an internal recurrence, and for each cell, there are the same inputs and outputs as the original data with different weights. Then, most of the critical information is utilized through the self-looping process, which

---

[2]See Appendix B.1 for details of FNN.

[3]See Section 2.2.4 for details of RNN.

improves the performance of predictability[4].

For predictions in time series, Hyndman and Athanasopoulos (2021) [51] summarize most of the classical methods, where different models have various structures. In my last chapter, I show that SSMs learn stock returns through forming latent variables, which are good at depicting trends and variations with great interpretability. Also, these models are manually selected and optimized for specific scenarios. While forecasting thousands of firm-level stock returns or building trading strategies, it can be hard to efficiently generate out-of-sample predictions. Moreover, considering selections of lagged variables, classical methods require closed-form solutions, which may reduce the efficiency of data.

To solve these problems, deep learning methods are introduced. In general, deep learning models relax assumptions about modeling, such as stationarity or heteroskedasticity, which widen the domain of applying these models (Rangapuram et al. (2018) [71])[5]. Also, they can be used to train thousands of time series simultaneously and discover similar patterns, even for those with little or no history. Further, considering the lack of interpretability, I use the family of probabilistic deep learning models to forecast aggregate market returns.

2.2.3 Probabilistic Deep Learning Models

Probabilistic time series forecasting estimates the conditional distribution of future time series given its past. Given strong structural assumptions, classical time series models are data efficient and can produce uncertainty forecasts (Wang et al. (2019) [92]). But, they "fail to capture complex patterns in the data, and multivariate techniques struggle to scale to large problem sizes" (Wang et al. (2019) [92]). On the other hand, deep neural networks can learn sophisticated patterns but rely on large data. By combining deep learning with classical models, researchers invented methodologies for producing precise probabilistic forecasts with benefits from both sides and less manual effort (Salinas et al. (2020) [83]).

Probabilistic deep learning models learn the entire dataset through a global model, where

---

[4]See Section 2.2.5 for more information of LSTM.

[5]Please see Section 2.3 for details.

similar patterns may exist across various time series. Then, using these models can help to predict thousands or millions of related time series with limited or no history available (Salinas et al. (2020) [83]). Also, based on the estimated conditional distribution of future time series, they can produce consistent quantile estimations (Rangapuram et al. (2018) [71])[6].

2.2.4 Recurrent Neural Network

I follow Goodfellow et al. (2016) [36] to give a brief introduction to recurrent neural networks (RNN), which "are a family of neural networks for processing sequential data." Unlike the feedforward neural network, recurrent neural networks use both feedforward and feedback connections to estimate or predict variables. While sharing parameters through evaluating process, it is possible for the model to utilize information from samples with different frequencies or lengths and generalize through layers.

A dynamical system driven by an external signal $x^{(t)}$ is given as:

$$h^{(t)} = f\left(h^{(t-1)}, x_t; \theta\right)$$

where $h^{(t)}$ denotes the hidden state of the system at step t, x_t denotes the inputs at step t, y_t denotes the target outputs at step t, and θ denotes the parameter of a mapping function f.

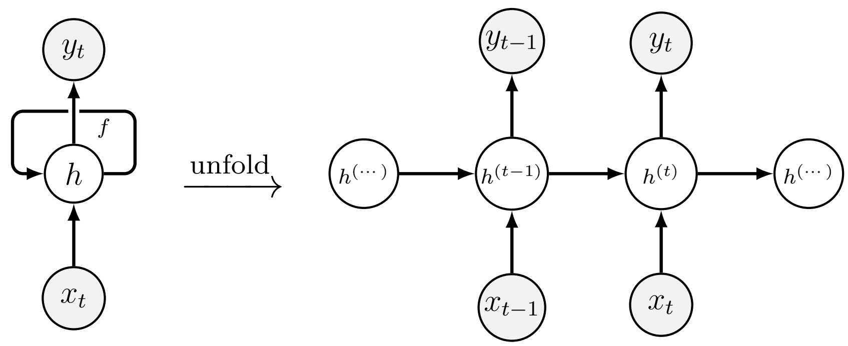
As shown in Figure 2.1, RNNs are non-linear dynamic systems with hidden states. The parameter used in RNNs can be used for different time steps, where fewer parameters are estimated than FNNs.

2.2.5 Long Short-Term Memory Model

Learning long-term dependencies in RNNs via gradient-optimization process can result in the ill-conditioned optimization problems, vanishing or exploding gradients. The long short-term

[6] Please see Section 2.3 for details.

Figure 2.1: Simple Recurrent Neural Network



This figure provides a diagram of simple recurrent neural networks (Left: A folded RNN; Right: An unfolded RNN). The recurrent neural network passes x_t to hidden states $h^{(t)}$, which are passed forward through time. y_t are the target outputs. The recurrent neural network has connections among inputs, hidden, and outputs, and each is parameterized by weight matrices.

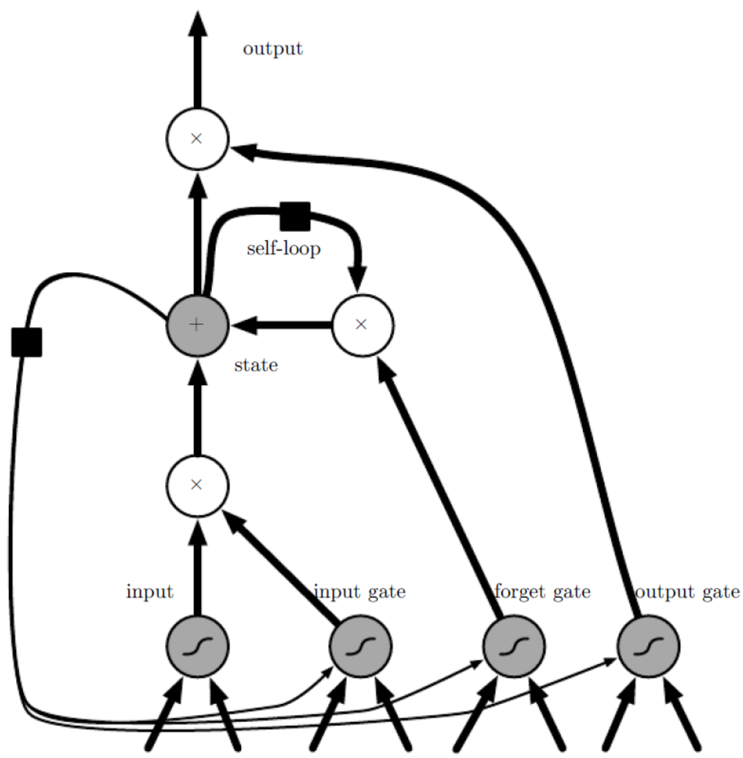
memory (LSTM) model (Hochreiter and Schmidhuber (1997) [46]) is one of the favorable solutions by introducing self-loops to produce paths that keeps gradients for long time.

As shown in Figure 2.2, by introducing self loops and forget gate, a RNN can produce input features within artificial neurons. Then, values from input gates can be accumulated into the state neurons, in which the weights are controlled by the forget gate and passed into the self loops. State cells can also generate inputs, and the outputs of cells can be controlled by the output gates. As a result, LSTM networks can improve the performance of estimating long-term dependencies through the self-looping process.

2.3 Models

This section briefly describes the deep learning methods that I use in my analysis. Starting from the first subsection, I introduce and describe a deep learning model in terms of three parts. First, I describe the contributions and intuitions behind each deep-learning forecasting model. Secondly, I briefly describe the process of estimating deep learning models, including the probabilistic function and structure of the algorithm. Finally, I discuss the tuning parameter sets for each deep-learning model, as they may have different parameter sets for

Figure 2.2: Block diagram of the LSTM recurrent network "cell"



This figure provides a block diagram of the LSTM recurrent network "cell", from Goodfellow et al. (2016) [36]. Cells are connected recurrently to each other, replacing the usual hidden units of ordinary RNNs. An input feature is computed with a regular artificial neuron unit. Its value can be accumulated into the state if the sigmoidal input gate allows it. The state unit has a linear self-loop whose weight is controlled by the forget gate. The output of the cell can be shut off by the output gate. All the gating units have a sigmoid nonlinearity, while the input unit can have any squashing nonlinearity. The state unit can also be used as an extra input to the gating units. The black square indicates a delay of a single time step.

various architectures.

All of the models are evaluated by the R^2, mean absolute prediction errors (MAE), and mean absolute percentage prediction errors (MAPE) for forecasting annual stock market returns. I discuss the trade-off between different evaluating metrics in the last subsection, where I also show the validating and tuning processes.

Though neural networks require fewer structural assumptions, they often leave the intuitions veiled. I focus my analysis on advanced techniques with more expressive power than standard neural nets. Probabilistic deep learning models estimate the conditional distribution of future time series, which can compute quantile estimates of asset returns. All the models are constructed based on LSTM or gated recurrent unit (GRU) RNNs, where the forget state takes care of the long-term flow of information without inducing vanishing or exploding problems.

2.3.1 Deep AR Model

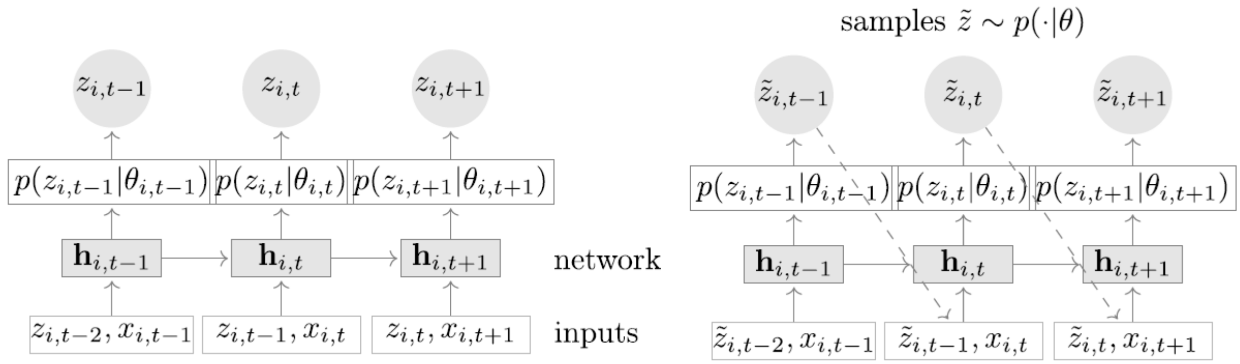
Following Salinas et al. (2020) [83], the deep AR model (DeepAR) is based on autoregressive recurrent neural networks, which predicts market returns by using a long short-term memory-based RNN architecture. Similar to the classical autoregressive model, DeepAR can capture seasonal behaviors and time dependencies in stock returns. It can automatically utilize the covariation across time series without further manual interventions. Compared with other deep learning models, DeepAR estimates the conditional distribution of future time series and provides more details of predicted values, including standard errors and confident interval, which helps to make optimal decisions.

Model. DeepAR focuses on forecasting one-dimensional (univariate) time series, and the training input is one or more target time series.[7] Unlike classical forecasting models, such as autoregressive integrated moving average (ARIMA) or exponential smoothing (ETS),

[7] Please see Section B.2 for examples of training data (model inputs).

DeepAR can fit multiple time series simultaneously instead of individually fitting each time series. Also, it can fit a time series with little or no history due to its algorithm.

Figure 2.3: Summary of the DeepAR Model



This figure provides a summary of the DeepAR model from Salinas et al. (2020) [83]. $z_{i,t}$ denotes the target value at time step t, $x_{i,t}$ denotes the covariates at time step t, $\mathbf{h}_{i,t}$ denotes the outputs from hidden states at time step t, and $\theta_{i,t}$ denotes the model parameters at time step t. Left: training model, at each time step t, inputs include covariates, lagged target values, and results from the previous hidden state. The output of hidden states at time step t, $\mathbf{h}_{i,t} = h(\mathbf{h}_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta)$, will be passed to the next hidden states and used to compute the parameter $\theta_{i,t} = \theta(\mathbf{h}_{i,t}, \Theta)$ of the likelihood function $p(z|\theta)$. Right: predicting model, at each time step t, predicted target values, $\tilde{z}_{i,t-1}$, along with covariates and outputs from previous states are used to predict target value in the next period. The predicting process keeps iterating until the end of the prediction range.

As shown in Figure 2.3, DeepAR follows the autoregressive recurrent neural network introduced by Graves (2013) [38] and Sutskever et al. (2014) [87]. For each time step t, DeepAR has inputs including covariates, $x_{i,t}$, lagged target values, $z_{i,t-1}$, and outputs from previous hidden states, $\mathbf{h}_{i,t-1}$. The goal is to model the conditional distribution of future time series:

$$P\left(\mathbf{z}_{i,t_0:T} \mid \mathbf{z}_{i,1:t_0-1}, \mathbf{x}_{i,1:T}\right) \tag{2.1}$$

where $\mathbf{z}_{i,t_0:T} \equiv [z_{i,t_0}, z_{i,t_0+1}, \cdots, z_{i,T}]$ denotes the future of time series, i, and $\mathbf{z}_{i,1:t_0-1} \equiv$

$[z_{i,1}, z_{i,2}, \cdots, z_{i,t_0}]$ denotes the history of the i-th time series.

Then, the distribution of DeepAR contains a product of likelihood factors:

$$\begin{aligned} Q_\Theta\left(\mathbf{z}_{i,t_0:T} \mid \mathbf{z}_{i,1:t_0-1}, \mathbf{x}_{i,1:T}\right) &= \prod_{t=t_0}^{T} Q_\Theta\left(z_{i,t} \mid \mathbf{z}_{i,1:t-1}, \mathbf{x}_{i,1:T}\right) \\ &= \prod_{t=t_0}^{T} p\left(z_{i,t} \mid \theta\left(\mathbf{h}_{i,t}, \Theta\right)\right), \end{aligned} \tag{2.2}$$

where $\mathbf{h}_{i,t} = h\left(\mathbf{h}_{i,t-1}, z_{i,t-1}, \mathbf{x}_{i,t}, \Theta\right)$ denotes the outputs from hidden network, which is parameterized by Θ through LSTM cells. Depending on the type of data, I incorporate Gaussian likelihood for market returns, where the likelihood function is parameterized by the mean and standard deviation, $\theta = (\mu, \sigma)$[8]. The mean is gained by applying the projecting function, and the standard deviation is transformed based on the softplus activation function[9]. Then, the likelihood function:

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log p\left(z_{i,t} \mid \theta\left(\mathbf{h}_{i,t}\right)\right) \tag{2.3}$$

is maximized directly to evaluate the parameters of hidden RNN, $h(\cdot)$, and distributions, $\theta(\cdot)$.

For the covariates, $x_{i,t}$ are assumed to be known for all the time series, and they can associate with individual time series, time frequencies, or both. For example, time-dependent covariates can be holidays or certain weeks of a year, including information on time patterns. Item-dependent can be shocks or news that affect the outcomes of target values. As I use annual stock data in this paper, there are no seasonality and covariates, $x_{i,t}$, do not exist.

[8]Functions $\mu(h_{i,t})$ and $\sigma(h_{i,t})$ map the outputs of the hidden state to the mean and covariance of a Gaussian distribution in this paper.

[9]Softplus activation function is $f(x) = \log(1+\exp(x))$, which forces the standard deviation to be positive

Tuning Process. Similar to general RNNs, I tune DeepAR based on the number of RNN layers, the number of RNN cells for each layer, batch size and learning rate[10].

Similar to RNNs, the number of layers control the depth of the DeepAR model, and the number of cells controls the evolution process in each layer. For the batch size and learning rate, they control the rate of convergence process for the DeepAR model. Instead of searching over uncountable parameter sets, I fix a range for each parameter to make reasonable comparisons across models. Details are described in section 2.3.5.

2.3.2 Deep State Space Model

Following Rangapuram et al. (2018) [71], the Deep State-Space Model (DSSM) forecasts asset returns by combining the classical state-space model (SSM) with recurrent neural networks. Traditional state-space models study the data by constructing a complex structure for each time series with a long enough history. Modelers have to specify all of the components that may involve with the targets. In other words, SSMs cannot infer patterns of data while evaluating nonlinear relationships, long-term dependencies, or multivariate time series.

As an alternative, deep neural networks require fewer structural assumptions and can effectively extract information from existing datasets. However, due to the structure of deep neural nets, they lack interpretability and generally require large datasets. Rangapuram et al. (2018) [71] propose the deep state-space model (DSSM) that parametrizes a state-space model by using recurrent neural networks for multivariate time series. The balance between interpretability and forecastability can improve the performance of predicting data without enough history and alleviate problems with overfitting.

Model. Similar to Fraccaro et al. (2017) [33], DSSM utilizes the highly efficient Kalman filter and uses a RNN to output SSM parameters directly, eliminating the additional tuning steps. DSSM focuses oon forecasting one-dimensional(univariate) time series with dynamic

[10] See gluonts.model.deepar for the full list of parameters

features, and the training input is one or more target time series. [11] As shown in Figure 2.4, let $z_{1:T_i}^{(i)} \in \mathbb{R}$ denote the i-th observed univariate time series, $\left(z_1^{(i)}, z_2^{(i)}, \ldots, z_{T_i}^{(i)}\right)$, ranging from 1 to T. $x_{1:T_i}^{(i)} \in \mathbb{R}^D$ denotes a set of time-varying covariate vectors, associating with target values, $z_{1:T_i}^{(i)}$. The goal is to model the distribution of future target values given the historical data:

$$p\left(z_{T_i+1:T_i+\tau}^{(i)} \mid z_{1:T_i}^{(i)}, \mathbf{x}_{1:T_i}^{(i)}; \Phi\right) \tag{2.4}$$

where Φ denotes the set of parameters of the model, shared across all the target time series data. Prediction range starts from the time period, $T+\tau$, and the training range is $\{1, 2, \cdots, T\}$. DSSM assumes that each target time series (an individual asset return), $\{z_{1:T_i}^{(i)}\}_{i=1}^{N}$, is independent of others, but due to the structure of RNN, it can always model the covariation among different time series through the sharing parameter, Φ.

By denoting $\Theta_t^{(i)}$ as the parameter set of a linear state-space model for the i-th time series, DSSM maps the covariate vectors, $x_{1:T_i}^{(i)}$,and target time series, $z_{1:T_i}^{(i)}$, to parameters of the state space model at time step t, $\Theta_t^{(i)}$. Then, this mapping is given as:

$$\Theta_t^{(i)} = \Psi\left(\mathrm{x}_{1:t}^{(i)}, \Phi\right), \quad i = 1, \ldots, N, \quad t = 1, \ldots, T_i \tag{2.5}$$

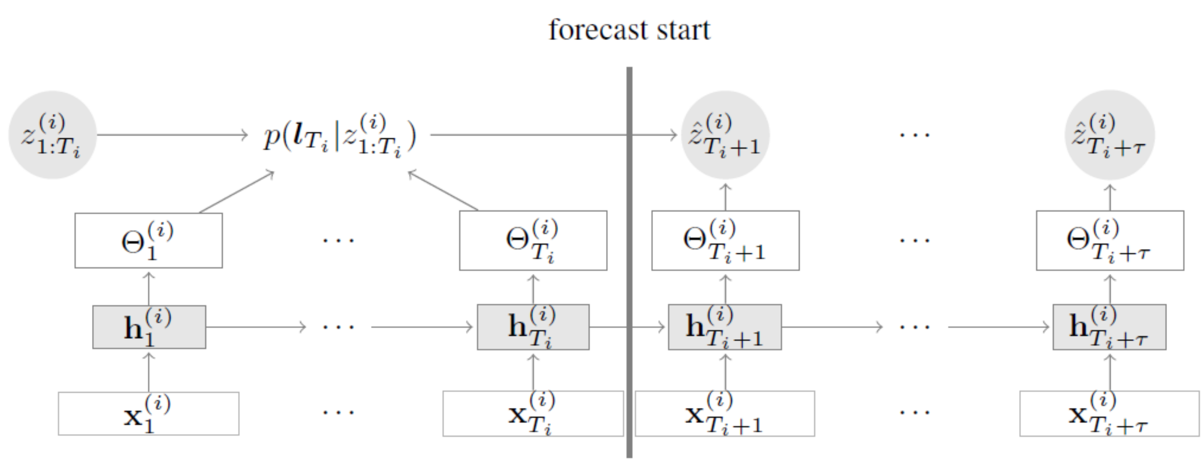
where Φ denotes the set of parameters of the hidden RNN, which is jointly learned from all the time series. And, the distribution of target time series is denoted as:

$$p\left(z_{1:T_i}^{(i)} \mid \mathrm{x}_{1:T_i}^{(i)}, \Phi\right) = p_{SS}\left(z_{1:T_i}^{(i)} \mid \Theta_{1:T_i}^{(i)}\right), \quad i = 1, \ldots, N \tag{2.6}$$

where p_{SS} denotes the marginal likelihood function for state-space models. Similar to the DeepAR model, after applying the affine transformation, results from hidden layers, $\mathbf{h}_t^{(i)} = h\left(\mathbf{h}_{t-1}^{(i)}, x_t^{(i)}, \Phi\right)$, are mapped into the parameters of state-space models, $\Theta_t^{(i)}$ (Rangapuram

[11] Please see Section B.2 for examples of training data (model inputs).

Figure 2.4: Summary of the Deep State-Space Model



This figure provides a summary of the Deep state-space model from Rangapuram et al. (2018) [71]. $z_t^{(i)}$ *denotes the target value at time step* t*,* $x_t^{(i)}$ *denotes the covariates associated with the i-th time series at time step* t*,* $\mathbf{h}_t^{(i)}$ *denotes the outputs from hidden states at time step* t*, and* $\Theta_t^{(i)}$ *denotes the parameters of the state space model at time step* t*.* $\Theta_t^{(i)}$ *are computed by maximizing the likelihood function of the state space model. Left: training model, given the historical target time series,* $z_t^{(i)}$*, and associated covariates,* $x_t^{(i)}$*, the posterior of the latent state,* l_t*, is formed as* $p(l_{T_i}|z^{(i)}_{1:T_i})$*. The output of hidden states at time step* t*,* $\mathbf{h}_{i,t} = h(\mathbf{h}_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta)$*, will be passed to the next hidden states and used to compute the parameter* $\theta_{i,t} = \theta\left(\mathbf{h}_{i,t}, \Theta\right)$ *of the likelihood function* $p(z|\theta)$*. Right: predicting model, starting at time period,* $T_i + 1$*, state-space parameters are obtained by the RNN model, and predictions fo target time series are generated through transition and measurement equations. The predicting process keeps iterating until the end of the prediction range.*

et al. (2018) [71]). By maximizing the log-likelihood function across different time series:

$$\mathcal{L}(\Phi) = \sum_{i=1}^{N} \log p\left(z_{1:T_i}^{(i)} \mid \mathbf{x}_{1:T_i}^{(i)}, \Phi\right) = \sum_{i=1}^{N} \log p_{SS}\left(z_{1:T_i}^{(i)} \mid \Theta_{1:T_i}^{(i)}\right) \tag{2.7}$$

we can get the optimal parameters, Φ^*, for the RNN model. During the process, DSSM automatically trains the state-space models and generate outputs without manual interventions.

Given Φ^*, probabilistic forecasts are made for each given time series. By assuming the multivariate Gaussian distribution, the joint distribution of predictions is estimated. Then, we can forecast market returns by recursively applying the latent system with outputs from the hidden RNN. This process also reduces the effort of tuning additional hyper-parameters of state-space models by using the outputs from RNN directly.

Different from the DeepAR model, DSSM utilizes target information through the likelihood term instead of using it as inputs directly. Also, DSSM is more computationally efficient than DeepAR due to the fewer times of unrolling RNN.

Tuning Process. Similar to DeepAR, I tune DSSM based on the number of RNN layers, the number of RNN cells for each layer, batch size, and learning rate[12]. Moreover, instead of searching over uncountable parameter sets, I fix a range for each parameter to make reasonable comparisons across models. Details are described in section 2.3.5.

2.3.3 Deep VAR Model

DeepVAR can be regarded as a multivariate variant of DeepAR. Following Salinas et al. (2019) [82], the Deep Vector Autoregressive Model (DeepVAR) focuses on forecasting multivariate time series by combining the Gaussian copula process output model with an LSTM-RNN. [13] In general, while researchers handle large datasets, they apply dimension reduction

[12] See gluonts.model.deepstate for the full list of parameters

[13] Please see Section B.2 for examples of training data (model inputs).

methods, such as principal component analysis, to datasets first and use classical vector autoregressive models (VAR) afterward to alleviate limitations on data. However, these models separate the steps of preprocessing and estimations, which restricts the learning process from the full dataset and can worsen the performance. Salinas et al. (2019) use a low-rank-plus-diagonal parametrization of the covariance matrix (See Figure 2.5) to reduce the computational complexity and number of parameters in the DeepVAR model.

Model. The goal for DeepVAR is to forecast the future values of the target time series by estimating the joint conditional distribution:

$$P(\mathbf{z}_{T+1}, \cdots, \mathbf{z}_{T+\tau} | \mathbf{z}_1, \cdots, \mathbf{z}_T) \tag{2.8}$$

where $z_1, \cdots, z_T$ denote the T observations, and $z_{T+1}, \cdots, z_{T+\tau}$ denote the future observations that need to be predicted. As shown in Figure 2.5, each time series is passed into a LSTM-RNN, and the joint distribution becomes:

$$p\left(\mathbf{z}_{T+1}, \ldots \mathbf{z}_{T+\tau} \mid \mathbf{z}_1, \ldots, \mathbf{z}_T\right) = p\left(\mathbf{z}_{T+1}, \ldots \mathbf{z}_{T+\tau} \mid \mathbf{h}_{T+1}\right) = \prod_{t=T+1}^{T+\tau} p\left(\mathbf{z}_t \mid \mathbf{h}_t\right) \tag{2.9}$$

Then, outputs of the hidden state, $h_{i,t}$, are passed into a Gaussian copula, which studies the structure of the dataset without assuming any prior marginal distribution for generating the data. As a result, following the Sklar's theorem, the log-likelihood function of the original observations is written as:

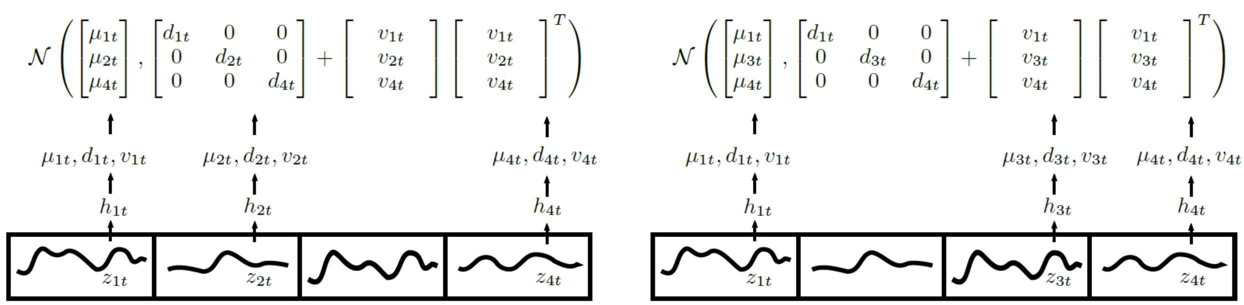
$$\log p(\mathbf{z}; \boldsymbol{\mu}, \Sigma) = \log \phi_{\boldsymbol{\mu},\Sigma}\left(\Phi^{-1}(\hat{F}(\mathbf{z}))\right) - \log \phi\left(\Phi^{-1}(\hat{F}(\mathbf{z}))\right) + \log \hat{F}'(\mathbf{z}). \tag{2.10}$$

where $\phi_{\boldsymbol{\mu},\Sigma}$[14] denotes the multivariate normal distribution, Φ denotes the cumulative density function (CDF) of the standard normal distribution, and $\hat{F}$ denotes the empirical CDF of

[14] Functions $\mu(h_t)$ and $\Sigma(h_t)$ map the outputs of the hidden state to the mean and covariance of a Gaussian distribution. See Figure 2.5 for an example.

the marginal distribution of chosen observations. Parameters of the distribution, $\boldsymbol{\mu}(\mathbf{h}_t)$ and $\boldsymbol{\Sigma}(\mathbf{h}_t)$, are obtained based on a low-rank-plus-diagonal parametrization of the covariance matrix, which reduces the computational complexity. By maximizing the summation of the log-density function across all the time steps, we can get the optimal model to forecast target values.

Figure 2.5: Summary of the Deep Vector Autoregressive Model



This figure provides a summary of the Deep VAR model from Salinas et al. (2019) [82]. $z_{i,t}$ denotes the target value at time step t, $\mathbf{h}_{i,t}$ denotes the outputs from the hidden states at time step t, and $\mu_{i,t}, d_{i,t}, v_{i,t}$ denotes the parameters of an assumed multivariate Gaussian distribution, which depend on the hidden-state outputs. These parameters are shared across all the time series DeepVAR model can be trained by considering a subset of time series in each batch, which is illustrated on the left (time series 1, 2, and 4) and right (time series 1, 3, and 4) parts.

DeepVAR is a multivariate variant of Deep AR, which links chosen time series by assuming a multivariate Gaussian distribution. DeepVAR reduces the number of parameters by incorporating a log-rank-plus-diagonal covariance matrix. Moreover, DeepVAR randomly samples a subset of time series for each batch, which alleviates the computational complexity while having a high-dimensional time series dataset. Following Salinas et al. (2019) [82], DeepAR and DeepVAR show similar experimental results, and both of them are representations of state-of-art in deep-learning forecasting. This paper shows similar results while predicting market returns.

Tuning Process. Similar to DeepAR, I tune DeepVAR based on the number of RNN layers, the number of RNN cells for each layer, batch size, and learning rate[15]. Though there are some additional tuning parameters, I simply fix a range for similar parameters in the DeepAR model to make reasonable comparisons across models. For the dataset used in this paper, the highest dimension is five, which is one of the reasons for fixing some parameters, such as the level of rank. Details are described in section 2.3.5.

2.3.4 Deep Factor Model

Following Wang et al. (2019) [92], this subsection introduces the Deep Factor model with a noise RNN (DF-RNN), which combines latent, global and deep components together. Considering the dependencies among time series, DF-RNN "represents each time series, or its latent function, as a combination of a global time series and a corresponding local model." By introducing the local model, DF-RNN captures random effects in each time series based on the chosen time series model. Similar to DeepAR, DF-RNN is a state-of-art deep-learning model for forecasting time series, and DF-RNN can be transformed into the DeepAR with restrictions on factors and random effects.

Model. Similarly to DeepAR model, DF-RNN focuses on forecasting one-dimensional (univariate) time series, and the training input is one or more target time series.[16] For DF-RNN model, we have inputs including covariates, $\boldsymbol{x}_{i,t} \in R^d$ and target values, $z_{i,t} \in R$, where $i \in \{1, 2, \cdots, N\}$ and $t \in \{1, 2, \cdots, T\}$. The goal is to calculate the conditional joint distribution of future time series:

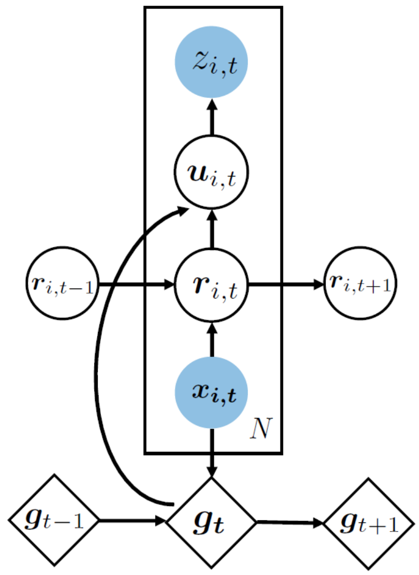
$$p\left(\{z_{i,T+1:T+\tau}\}_{i=1}^{N} \mid \{\boldsymbol{x}_{i,1:T+\tau}, z_{i,1:T}\}_{i=1}^{N}\right) \tag{2.11}$$

[15] See gluonts.model.deepvar for the full list of parameters

[16] Please see Section B.2 for examples of training data (model inputs).

where τ denotes the forecast horizon. Similar to DeepAR, covariates, $\boldsymbol{x}_{i,t}$, are assumed to be known for all the time series and associated with target values.

Figure 2.6: Summary of the Deep Factor Model



This figure provides a summary of the Deep Factor model, from Wang et al. (2019) [92]. $z_{i,t}$ denotes the target value at time step t, $\mathbf{x}_{i,t}$ denotes the covariates, g_t denotes the global effects based on the common patterns of all the time series, $r_{i,t}$ denotes the random effect for local fluctuations (individual time series), and $u_{i,t}$ denotes the values of latent functions. Then, we can form the conditional distribution of $z_{i,t}$ as $p\left(z_{i,t} \mid u_i\left(\boldsymbol{x}_{i,t}\right)\right)$.

Following Wang et al. (2019) [92], DF-RNN model has the following structure:

$$\begin{aligned}
\text{global factors:} \quad & g_k(\boldsymbol{x}_{i,t}) = \mathrm{RNN}_k(\boldsymbol{x}_{i,t}), \quad k = 1, \cdots, K, \\
\text{fixed effect:} \quad & f_i(\boldsymbol{x}_{i,t}) = \sum_{k=1}^{K} w_{i,k} \cdot g_k(\boldsymbol{x}_{i,t}), \\
\text{random effect:} \quad & r_i(\boldsymbol{x}_{i,t}) \sim \mathcal{N}\left(0, \sigma_{i,t}^2\right), \quad i = 1, \cdots, N, \\
\text{latent function:} \quad & u_i(\boldsymbol{x}_{i,t}) = f_i(\boldsymbol{x}_{i,t}) + r_i(\boldsymbol{x}_{i,t}), \\
\text{emission:} \quad & z_{i,t} \sim p\left(z_{i,t} \mid u_i\left(\boldsymbol{x}_{i,t}\right)\right)
\end{aligned} \tag{2.12}$$

As shown in Figure 2.6, the global effects are driven by the outputs of RNNs with a chosen number of latent factors, K. Then, effects from the latent dynamic system are restricted to be deterministic. Fixed effects use the outputs from global factors to customize the values for each time series through an embedding layer, $w_{i,k}$. For random effects, DF allows various probabilistic time series models and I choose a RNN to generate the variance of noises, where the standard deviation of $\varepsilon_{i,t}$ is denoted as $\sigma_{i,t} \equiv RNN(\boldsymbol{x}_{i,t})$. As a result, we have the likelihood function as:

$$p\left(\boldsymbol{z}_i\right) = \prod_t \mathcal{N}\left(z_{i,t} - f_{i,t} \mid 0, \sigma_{i,t}^2\right) \tag{2.13}$$

Then, by maximizing the log-likelihood function across different time series, we can get the optimized parameter set for local and global models.

DF-RNN can be transformed into a DeepAR model by having one factor and no random effects and by adding autoregressive inputs (Wang et al. (2019) [92]). The difference comes from the scaling of data, where DF-RNN automatically scale each time series instead of pre-select. Moreover, by adding random effects, DF-RNN generally performs well for forecasting multiple time series with different local patterns. With more time-series data, DF-RNN is slightly better than other deep-learning methods (see details in section 2.4)

Tuning Process. DF-RNN has a different structure from the previous models, and the tuning process is more complex. I tune DF-RNN based on the number of hidden layers for both global and local RNN models, the number of units per hidden layer for both global and local RNN models, the number of global factors, batch size, and learning rate[17]. The deep factor RNN model asks for the number of factors for the dynamic system, which is related with the dimension of data. Details are described in section 2.3.5.

[17]See gluonts.model.deep_factor for the full list of parameters

2.3.5 Validating and Tuning Processes

To validate and select the models with best performance, I follow the standard way to split observations into training and testing sets. For time series or sequential data, if we randomly split the data, then the information existing in the testing dataset may lead to bias because of data-leakage problems and may destroy existing dependencies. Thus, I form the training dataset (90%) containing the data from 1950 to 2012, and the testing dataset (10%) containing the data from 2013 to 2019. Moreover, to robustly test models, I generate a second training dataset containing 85% of the data, and the second testing dataset includes the rest.

Performance Evaluations. To compare the performance among various models, I use out-of-sample R^2_{Ret}, Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) as the evaluating stats. Same as the last chapter, R^2_{Ret} is defined as:

$$R^2_{\text{Ret}} = 1 - \frac{v\hat{a}r\left(r_{t+1} - \hat{r}_{t+1}\right)}{v\hat{a}r\left(r_t\right)} \tag{2.14}$$

where $v\hat{a}r$ denotes the sample variance, and $\hat{r}_{t+1}$ are the predictions of market returns (Van Binsbergen and Koijen (2010) [88]). MAE is defined as:

$$\text{MAE} = \frac{1}{T}\sum_t |r_{t+1} - \hat{r}_{t+1}| \tag{2.15}$$

MAPE is defined as:

$$\text{MAPE} = \frac{1}{T}\sum_t |\frac{r_{t+1} - \hat{r}_{t+1}}{r_{t+1}}| \tag{2.16}$$

R^2 shows the wellness of explaining variability in the model, and can be largely affected by outliers. MAPE[18] is skewed towards the condition that actual returns are small (r_{t+1}

[18] If r_{t+1} approximates to zero, a noise, ε, is added to prevent from exploding.

is small), which is useful when values significantly differ across time. And MAE shows the average of absolute residuals, which is robust to outliers. I present these three metrics to understand the empirical results thoroughly.

Also, due to the randomness of predictions, I generate one-hundred sample paths for each model, and take the mean of sample paths to produce performing metrics. It is also possible to create empirical quantile loss, which is robust to outliers and over- or under-fittings. I will further show empirical results in the third chapter.

Hyper-parameter Tuning. To forecast market returns, I conduct a grid-search within the chosen range of hyperparameters to find the best values. For each set of hyperparameters, I fit the model on the training set and generate evaluation metrics on the testing set. With small datasets, neural networks often perform well with a few layers and cells. For each model, the dimensionality of inputs is different, and the controlling parameter is also changed. Table 2.1 lists the parameters that are tuned and the searching range. Other parameters are set as default.

Table 2.1: Hyper-parameters Values Range Searched in Hyper-parameter Tuning

	DeepAR, DSSM, DeepVAR	DF-RNN
No. of RNN layers	1,2,3,4,5	-
No. of RNN cells per layer	8,16,32	-
Batch size	16,32,48	16,32,48
Learning rate	0.1,0.01,0.001,0.0001	0.01,0.001,0.0001
No. of units per layer for global RNN	-	25,50
No. of layers for global RNN	-	1,2,3,4,5
No. of global factors	-	1,2,3,4,5,6
No. of units per layer for local RNN	-	16,32,48
No. of layers for local RNN	-	1,2,3,4,5

2.4 *Empirical Findings in U.S. Stock Market*

I implement deep-learning models using GluonTS (Alexandrov et al. (2020) [1]) and use AWS SageMaker to forecast U.S. market returns. This paper uses the same datasets (annual aggregate stock market returns associated with annual dividend-price ratios, dividend growths, returns on equity, and book-to-market ratios) as the first chapter to directly compare deep-learning methods with classical time series models from the first chapter. Train-test splitting follows the procedure described in Section 2.3.5. The length of the prediction horizon is eight.

This section shows results from various models with different datasets and discusses the differences among experiments. Section 2.4.1 describes the data, and Section 2.4.3 presents the results from other experiments. Comparisons of experiments are shown in Section 2.4.4.

2.4.1 Data

This chapter uses the same data set as the first chapter.

Manipulations. By utilizing the value-weighted stock market returns from CRSP, I generate the price-dividend ratio and dividend growth ratio. For firm-level data from COMPUSTAT, firms must have December as the fiscal-year end to align accounting variables across firms. To filter out data errors, I exclude firms with less than \$10 million market values and more than 100 or less than a 0.01 book-to-market ratio.

Firm-level variables are calculated as follows. The market value of equity is the product of common outstanding shares and the closing price in certain fiscal years. For book equity, I use the total common equity (data item 60). If the data is not available, I use the liquidity value (data item 235) as a substitute. Considering taxes, short-term and/or long-term deferred taxes (data item 35 and 71) are added to book equity if available. If neither the total common equity nor the liquidity value is available, I use the clean surplus identity (Equation (1.22)) to approximate book value. All firm-level book equity must be non-negative to be included in the analysis.

Firms' net incomes (data item 172) are regarded as earnings, and if the data is missing, earnings are approximated using the clean surplus identity (Equation (1.22)). Return on equity (ROE) or profitability is the earnings over the last period's book equity. Intuitively, firms cannot lose more than their book values. Thus, if firms have negative earnings, the absolute value of earnings must be smaller than its book equity.

To convert firm-level data into aggregate-level data, I utilize market capitalization to calculate value-weighted variables. Market-level data are calculated as the value-weighted mean of existing variables scaled by the price level in the fiscal year.

2.4.2 Model Inputs

To equally compare results, I use three different datasets. The first one, the "stock-variable dataset", contains market returns, dividend growths, and dividend-price ratios. The second one, the "accounting-variable dataset", contains market returns, returns on equity, and book-to-market ratios. The third one has all the time series in the first and second datasets.

DeepAR The data layout of DeepAR is univariate (Salinas et al. (2020) [83]).[19] In other words, each time series is represented as an individual line following the JSON format. *DeepAR-r* denotes the DeepAR model with only one input, annual market returns. I denote the DeepAR model with stock or accounting variables as, *DeepAR-s* and *DeepAR-a*, where they both have three time series. *DeepAR-all* denotes the DeepAR model with all the data, five time series. This model forecasts all the target inputs (all the individual/market asset returns) at the same time.

DSSM The data layout of DSSM is univariate (Rangapuram et al. (2018) [71]).[20] In other words, each time series is represented as an individual line associated with dynamic features, such as dividend growth, following the JSON format. Also, to compare with classical

[19]Please see Section B.2 for examples of training data (model inputs).

[20]Please see Section B.2 for examples of training data (model inputs).

SSM, DSSM takes lags of dynamic features as inputs. *DSSM-s* denotes the DSSM model with stock variables, and dividend growths and dividend-price ratios are used as dynamic features. *DSSM-a* denotes the DSSM model with accounting variables, and returns on equity and book-to-market ratios are used as dynamic features. *DSSM-all* denotes the DSSM model with all the data, where market return is the target time series associated with all the other data. This model forecasts only the target time series, market returns. [21]

DeepVAR The data layout of DeepVAR is multivariate (Salinas et al. (2019) [82]).[22] In other words, both target time series and features are represented in the same line following the JSON format. Also, to compare with classical methods, DeepVAR takes lags of features as inputs. *DeepVAR-s* denotes the DeepVAR model with stock variables. *DeepVAR-a* denotes the DeepVAR model with accounting variables. *DeepVAR-all* denotes the DeepVAR model with all the data. This model forecasts all the available inputs at the same time, including market returns, dividend growth rates and others.

DF-RNN The data layout of DF-RNN is univariate (Wang et al. (2019) [92]).[23] In other words, each time series, such as market returns and dividends, is represented as an individual line following the JSON format. I denote the DF-RNN model with stock or accounting variables as, *DF-RNN-s* and *DF-RNN-a*, where they both have three time series. *DF-RNN-all* denotes the DF-RNN model with all the data, five time series. This model forecasts all the available inputs at the same time, including market returns, dividend growth rates and others.

---

[21] DSSM does not forecast dynamic features of market returns, such as dividend growth rates.

[22] Please see Section B.2 for examples of training data (model inputs).

[23] Please see Section B.2 for examples of training data (model inputs).

2.4.3 Results

In this section, I show results from linear regressions and probabilistic deep learning models. Table 2.2 shows parameter estimates of linear regressions with different datasets. Based on the testing R^2, we can see that simply increasing lags may weaken the model performance.

Table 2.2: Results from Linear Regressions (90/10 Split)

	r_t	r_t	r_t	r_t	r_t	r_t	r_t	r_t
const	0.0698	0.0861	0.5120	-0.0455	0.1114	0.1459	0.7006	-0.1920
	(0.0238)	(0.0227)	(0.1942)	(1.5827)	(0.0376)	(0.0463)	(0.3205)	(1.4297)
r_{t-1}	-0.0682	-0.0860	-0.2333	3.6156	-0.0228	0.1773	-0.2389	9.2570
	(0.0915)	(0.1110)	(0.1279)	(11.2007)	(0.0967)	(0.1994)	(0.1139)	(11.0155)
dg_{t-1}	-	-	0.3345	-3.6441	-	-	0.3104	-9.2000
	-	-	(0.1032)	(11.2196)	-	-	(0.1078)	(11.0318)
dp_{t-1}	-	-	0.1277	3.7229	-	-	0.1999	9.0770
	-	-	(0.0582)	(10.8329)	-	-	(0.1118)	(10.7230)
e_{t-1}	-	-	-	-	0.1606	0.9123	0.4447	1.6690
	-	-	-	-	(0.4282)	(0.1232)	(0.5199)	(0.6235)
b/m_{t-1}	-	-	-	-	0.1371	0.4096	-0.0412	0.2030
	-	-	-	-	(0.0560)	(0.2168)	(0.1127)	(0.2507)
r_{t-2}	-	-0.2403	-	-0.1062	-	-0.2373	-	-0.2010
	-	(0.0670)	-	(0.1874)	-	(0.0726)	-	(0.1626)
dg_{t-2}	-	-	-	-0.1863	-	-	-	-0.2380
	-	-	-	(0.2716)	-	-	-	(0.2232)
dp_{t-2}	-	-	-	-3.7227	-	-	-	-9.0300
	-	-	-	(11.1800)	-	-	-	(11.0047)
e_{t-2}	-	-	-	-	-	-1.4776	-	-1.4180
	-	-	-	-	-	(0.4736)	-	(0.4978)
b/m_{t-2}	-	-	-	-	-	-0.3494	-	-0.4140
	-	-	-	-	-	(0.2325)	-	(0.2441)
Test R^2	0.0372	0.2439	0.2641	0.3262	0.0734	-1.0964	0.3270	-1.1598

This table shows parameter estimates for linear regressions of market returns. r denotes log market returns, dp denotes log dividend-price ratio, dg denotes log dividend growth rate, e denotes log returns on equity, and b/m denotes book-to-market ratio. The models are estimated based on the sample data from 1950 - 2019, CRSP and COMPUSTAT. Train-test split ratio is 90:10.

Table 2.3 shows performance metrics for all the models mentioned above. Compared with Table 2.2, market returns have much higher testing R^2. Panel A shows the results from DeepAR model with only one input, log market returns. Comparing to the classical SSM with stock variables, the R^2 raises from 0.3142 to 0.3678. Given the definition of R^2, we can see that the out-of-sample predictions are close to the observed data, as shown in Figure 2.7a.

Table 2.3: Results from Deep Learning Models (90/10 Split)

	Dataset	MAE	MAPE	R^2
Panel A: Forecast of Market Returns with Univariate Input				
DeepAR-r	r	0.1444	0.6584	0.3678
Panel B: Forecast of Market Returns with Stock Variables				
DeepAR-s	r, dp, dg	**0.1037**	0.6986	**0.4233**
DSSM-s	r, dp, dg	0.22791	2.0138	0.3075
DeepVAR-s	r, dp, dg	0.1257	**0.53970**	0.3716
DF-RNN-s	r, dp, dg	0.1467	0.9937	0.1642
Panel C: Forecast of Market Returns with Accounting Variables				
DeepAR-a	r, e, b/m	0.1197	1.0677	0.4857
DSSM-a	r, e, b/m	0.1304	**0.7141**	**0.5092**
DeepVAR-a	r, e, b/m	0.1215	0.8165	0.2876
DF-RNN-a	r, e, b/m	**0.1045**	0.8978	0.2986
Panel D: Forecast of Market Returns with All Variables				
DeepAR-all	r, dp, dg, e, b/m	0.1316	1.0506	**0.6580**
DSSM-all	r, dp, dg, e, b/m	0.1363	**0.5529**	0.4964
DeepVAR-all	r, dp, dg, e, b/m	**0.1018**	0.6483	0.4245
DF-RNN-all	r, dp, dg, e, b/m	0.1949	1.4176	0.6279

This table shows testing statistics for out-of-sample predictions of market returns. The second column presents the information set for each model. r denotes log market returns, dp denotes log dividend-price ratio, dg denotes log dividend growth rate, e denotes log returns on equity, and b/m denotes book-to-market ratio. The best results are marked in bold (R^2: the higher is better; MAE: the lower is better; MAPE: the lower is better.) The models are estimated based on the sample data from 1950 - 2019, CRSP and COMPUSTAT. Train-test split ratio is 90:10.

In Panel B, I show the results of all deep learning models with stock variables (dividend-price ratios & dividend growth rates). Given different metrics, DeepAR has the smallest MAE and highest R^2, while DeepVAR has the smallest MAPE. As shown in Figure 2.7b, DeepAR fits the data better than others, and captures almost all market returns movements in the testing dataset. DeepVAR fits the model With a higher MAPE than DeepAR when observations are relatively small. With stock variables, DSSM generates similar out-of-sample prediction performance as in the first chapter, and roughly showing the movements of real market returns.

Comparing results from DeepAR in Panel B with those in Panel A shows that including more time series or information helps to improve the performance. The hidden RNN is trained better than only using univariate time series, as it can capture covariates among various time series.

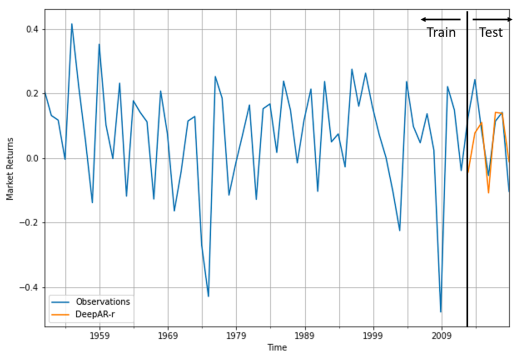
In Panel C, I present the results based on using accounting variables (book-to-market ratios & returns on equity). DSSM has the highest R^2 and lowest MAPE, and DF-RNN has the smallest MAE with a smaller R^2. As shown in Figure 2.7c, DSSM captures all the volatility but with a smaller magnitude than observations. DF-RNN does not strictly follow the trend, but it has smaller predictions than others, which induces a smaller MAE. DeepAR also has similar R^2 and MAE to DSSM, but it deviates from observed values.

Comparing the results of DSSM in Panel C (50.92% R^2) to the ones from chapter one (35.62% R^2) and Panel A, we can see that with accounting variables, DSSM can capture more information and further support the informative power of accounting variables. Also, lack of essential factors in the accounting identity may weaken the performance in the classical SSM, as we may not have enough knowledge of the latent dynamic system.

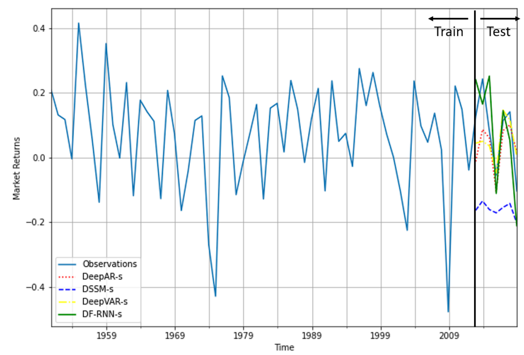
Given the results in Panel A and B, I conclude that accounting data are more relevant to aggregate market returns than stock variables. As dividend policies are unstable over time, accounting data can be used as substitutes to forecast market returns.

In Panel D, I summarize the results using all the data (dividend-price ratios, dividend growth rates, book-to-market ratios, and returns on equity). Also, it gives the best results

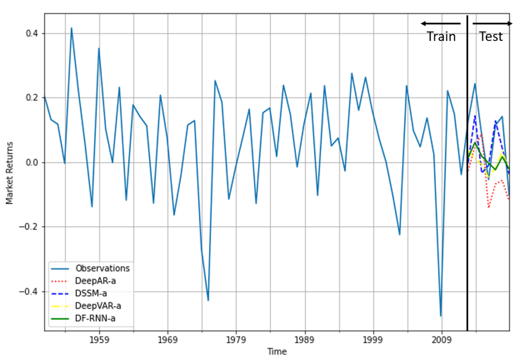
Figure 2.7: Out-of-sample Predictions of Market Returns with Different Models and Variable Sets



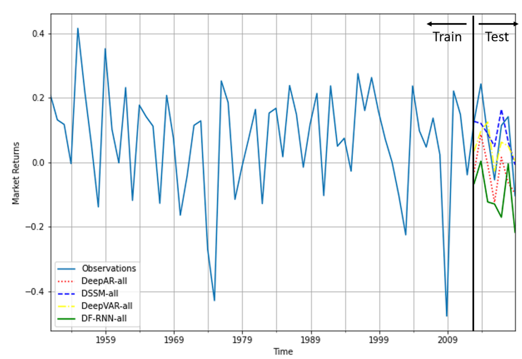
(a) DeepAR w/ Univariate Market Returns



(b) Deep Learning w/ Stock Variables



(c) Deep Learning w/ Accounting Variables



(d) Deep Learning w/ All Variables

This plot shows four subplots that predicts out-of-sample market returns. Part (a) is generated by DeepAR model with univariate time series, market returns. Part (b) and (c) use stock or accounting variables to generate out-of-sample predictions of market returns based on four different deep learning models. Part (d) shows out-of-sample predictions of market returns for various deep learning models. The models are estimated based on the sample data from 1950 - 2012 and tested on sample data from 2013 - 2019, CRSP and COMPUSTAT. Train-test split ratio is 90:10.

compared with previous ones. DeepAR has the highest R^2; DeepVAR has the smallest MAE; and DSSM has the smallest MAPE. As shown in Figure 2.7d, DeepAR precisely captures

Table 2.4: Results from Deep Learning Models (85/15 Split)

	Dataset	MAE	MAPE	R^2
Panel A: Forecast of Market Returns with Univariate Input				
DeepAR-r	r	0.1071	0.6154	0.3141
Panel B: Forecast of Market Returns with Stock Variables				
DeepAR-s	r, dp, dg	0.1306	**0.7231**	**0.4223**
DSSM-s	r, dp, dg	0.1120	1.1207	0.3126
DeepVAR-s	r, dp, dg	**0.0951**	0.8549	0.4201
DF-RNN-s	r, dp, dg	0.1530	0.9467	0.3497
Panel C: Forecast of Market Returns with Accounting Variables				
DeepAR-a	r, e, b/m	**0.1047**	0.8363	**0.4469**
DSSM-a	r, e, b/m	0.1254	**0.7137**	0.3617
DeepVAR-a	r, e, b/m	0.1124	0.7908	0.4197
DF-RNN-a	r, e, b/m	0.1344	0.8533	0.3603
Panel D: Forecast of Market Returns with All Variables				
DeepAR-all	r, dp, dg, e, b/m	0.2467	2.1643	0.5200
DSSM-all	r, dp, dg, e, b/m	0.1558	0.6785	0.4491
DeepVAR-all	r, dp, dg, e, b/m	**0.1017**	0.7136	0.5140
DF-RNN-all	r, dp, dg, e, b/m	0.1711	**0.5259**	**0.5793**

This table shows testing statistics for out-of-sample predictions of market returns. The second column presents the information set for each model. r denotes log market returns, dp denotes log dividend-price ratio, dg denotes log dividend growth rate, e denotes log returns on equity, and b/m denotes book-to-market ratio. The best results are marked in bold (R^2: the higher is better; MAE: the lower is better; MAPE: the lower is better.) The models are estimated based on the sample data from 1950 - 2019, CRSP and COMPUSTAT. Train-test split ratio is 85:15.

the movements of market returns, but with smaller magnitudes. DF-RNN shows similar trends, but it deviates farther than DeepAR. DSSM shows similar results as in Panel C, but it forecasts market returns accurately when market returns are small (smaller MAPE).

Using all the available data, these models generally perform better than previous ones, especially for DF-RNN. DF-RNN is different from other models by having both global and local RNNs, which may require a larger dataset than other models. Moreover, with more complex structures require more data to efficiently forecast market returns.

Further, to robustly check the performance of these models, I present the results based on the second training (85%) and testing (15%) datasets in Table 2.4. Comparing with Table 2.3, there are similar results, but the best model's performance is slightly weaker due to fewer observations in the training dataset. With more informative variables, predictions are more accurate, as shown in Table 2.4 Panel D. It also supports the argument that accounting data is more informative than stock variables, as the metrics in Panel C are slightly better than the ones in Panel B. Figure 2.8 shows the forecasting path of market returns based on the second training and testing datasets.

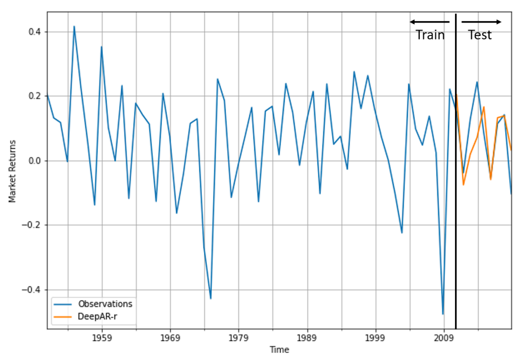
2.4.4 Discussions

In this section, I briefly compare deep learning models for predicting stock market returns based on model requirements, algorithms of learning data, and results from the previous section.

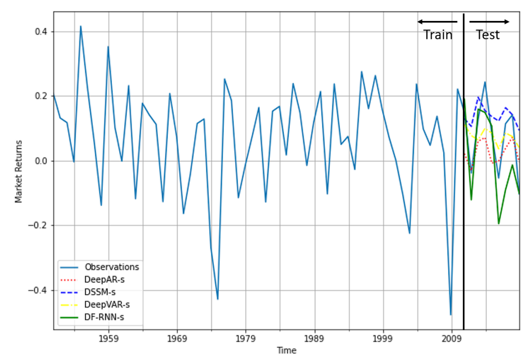
As mentioned above, these models have different ways of inputting data. DeepAR and DF-RNN require a univariate layout of data, which means that all the data have to be imported as individual time series. Though inputs may contain covariates, they must be known all the time, such as a specific day of a week. DF-RNN utilizes information by building local and global RNNs, where common patterns are shared through the global RNN. Similarly, DeepAR uses a RNN to learn covariation among these data and make predictions directly. Also, DF-RNN can be transformed into DeepAR by restricting the number of factors to one, eliminating random effects, and adding autoregressive inputs. Due to the complex structure of DF-RNN, it performs better with larger datasets.

Unlike DeepAR, DSSM forecasts each target time series with dynamic features instead of lagged values, and it only uses target variables through the likelihood function. With informative variables, DSSM performs better than DeepAR, as shown in Table 2.3, Panel A and B. However, DeepAR performs better than DSSM when the data size gets larger because

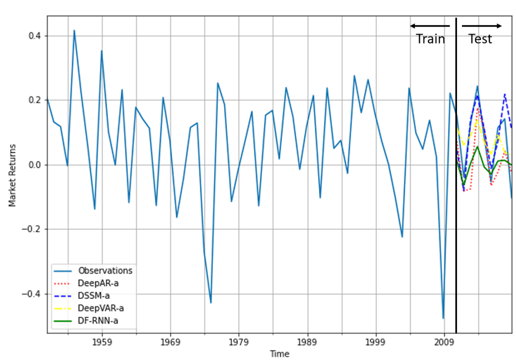
Figure 2.8: Out-of-sample Predictions of Market Returns with Different Models and Variable Sets



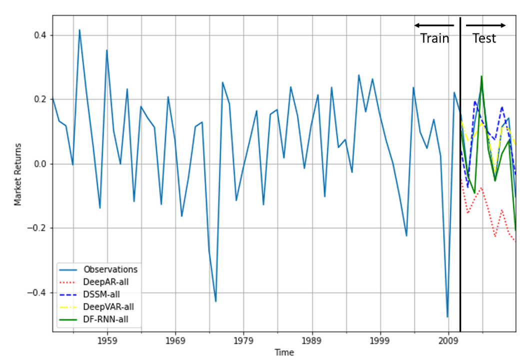
(a) DeepAR w/ Univariate Market Returns



(b) Deep Learning w/ Stock Variables



(c) Deep Learning w/ Accounting Variables



(d) Deep Learning w/ All Variables

This plot shows four subplots that predicts out-of-sample market returns. Part (a) is generated by DeepAR model with univariate time series, market returns. Part (b) and (c) use stock or accounting variables to generate out-of-sample predictions of market returns based on four different deep learning models. Part (d) shows out-of-sample predictions of market returns for various deep learning models. The models are estimated based on the sample data from 1950 - 2012 and tested on sample data from 2013 - 2019, CRSP and COMPUSTAT. Train-test split ratio is 85:15.

adding more dynamic features in DSSM is less effective than having more covariation in DeepAR (See Table 2.3 Panel D).

Compared with traditional SSMs, DSSM avoids assuming the incorrect or inaccurate structure of the latent dynamic system. DSSM does not require a specific form of the dynamic system and automatically trains a globally shared RNN to generate inputs of state-space models. Due to the process, DSSM does not require a long history of the data. Compared with DeepAR, DSSM is more robust to noise or outliers, as it only includes target values through the likelihood term.

Similar to DeepAR, DeepVAR requires a multivariate layout of data, where all the time series are treated as target variables. DeepVAR automatically reduces the computational complexity and number of evaluated parameters, which is less time-consuming but performs a little worse than others when the data size is small.

In general, DeepAR, DSSM, and DF-RNN can predict time series data with little or no history available, but DeepVAR requires historical data to make predictions. Moreover, due to the complexity of structures, DeepVAR and DF-RNN require a larger dataset to perform well compared with DeepAR and DSSM. While forecasting aggregate U.S. market returns, DF-RNN and DeepAR perform best with all the available data (shown in section 2.4.3). However, while predicting firm-specific stock returns, there may be thousands of firms, and the data length may differ significantly. As DF-RNN and DeepAR cannot import features for each individual firm, such as dividends and earnings, it may affect their performance. At the same time, when we have new-listing companies in the dataset, it may be a problem for DeepVAR due to its historical data requirements. To further understand the differences among these models, I test them in the third chapter.

2.5 Conclusion

For this chapter, I test state-of-art models based on the architecture of probabilistic recurrent-neural-network with long short-term memory. I use the same datasets with stock and accounting variables to compare equally with the first chapter. Results from deep learning models further support the statement in the first chapter that accounting variables are more informative for forecasting market returns.

Also, based on results from Table 2.3, I conclude that probabilistic deep-learning models outperform traditional methods by presenting R^2 ranging from 42.33% to 65.80%. These models utilize neural nets' computational advantages without largely reducing the interpretability. Moreover, considering the history of data, we have to choose ideal models based on different scenarios. DSSM and DeepAR perform better while having fewer inputs. DeepVAR's and DF-RNN's performance increases with the size of datasets.

Moreover, the deep state-space model is a straightforward improvement of traditional state-space models, which keeps the highly efficient Kalman filter and incorporates RNNs to improve the performance. By automatically studying the outputs of RNNs, DSSM also reduces the effort of building sophisticated transitional system and avoid inaccurate knowledge of latent variables. Results of DSSMs in Table 2.3 show that the predictive system of classical SSM with stock variables sufficiently discovers existing information, as the performance of out-of-sample predictions is similar to the one from the first chapter. But, with accounting variables, DSSM outperforms the traditional SSM by having R^2 equal to 50.92%.

To thoroughly examine the performance of deep learning methods, I will train models on expanded information sets in the next chapter. In other words, I will include more features for the aggregate-level stock returns and estimate models with firm-level stock returns and features. Also, I will focus on specific sectors of the market, such as the housing market, to see if these models are stable while interacting with human decisions.

Chapter 3

PROBABILISTIC DEEP-LEARNING MODELS FOR PREDICTING U.S. STOCK RETURNS

3.1 Introduction

In this chapter, I perform a comparative analysis of probabilistic deep learning models by forecasting U.S. stock returns at both aggregate and firm levels. My primary contributions are tri-fold. First, I use larger datasets to forecast aggregate-level stock returns based on different train-test datasets. With ten features, expected U.S. market returns have out-of-sample R^2 varies ranging from 58.55% to 76.51% based on the 90-10 train-test split, which outperforms the results from previous chapters. Also, with an 85-15 train-test split, the out-of-sample R^2 ranges from 44.19% to 63.14%. These results also reinforce the result from the second chapter that probabilistic deep learning models can outperform traditional time series models, such as vector autoregressive and state-space models, in forecasting aggregate stock retuirns.

Second, I use a large dataset with 102 features to forecast firm-level annual stock returns, where the average number of stocks per year is around 5,635. With a large dataset, missing values and noisy returns significantly deteriorate the performance of deep learning models, especially for those with univariate data layouts. I filter the dataset with a Truncated SVD to reduce the dimensionality and generate informative features. Here, I find that DSSM provides optimal predictions with a 4.52% R^2 on testing samples. I also split firm-level stock returns into groups based on market values, book-to-market ratios, and expected future returns. I find that middle-size stocks generally have more accurate predictions than others. Also, predictions of neutral and value stocks are more precise than growth stocks.

Given the predicted results above, I build a zero-net-investment strategy. Essentially, I build equally-weighted and value-weighted portfolios by buying and selling the same amounts of securities, and the net value of these positions is zero. After sorting stocks based on their expected returns into ten deciles, I long stocks with the highest expected returns and short those with the lowest ones. I find that the DSSM can build the optimal trading strategy with a 30% average annual excess return and more than 400% cumulative returns over the both testing (2008 - 2013) and out-of-sample periods (2014 - 2019). Though DeepAR provides a slightly larger excess return, its predictions are more volatile than DSSM, given its variance of predicted and real returns.

I find that the DSSM is more robust to noises than the other models and has the highest R^2 on the firm-level testing dataset. It also helps to build a profitable trading strategy with value and equally weighted portfolios. DeepAR is the optimal method with a univariate data layout while having a small dataset, and DF-RNN is better than others while having a large dataset. With informative enough features, DeepVAR can generate the best predictions.

The organization of this chapter is as follows. In Section 3.2, I describe datasets for aggregate and firm-level stock returns, along with the tuning processes. Section 3.3 shows forecasting results of aggregate stock returns based on different models. In section 3.4, I outline the empirical predictions based on U.S. firm-level stock returns and build a zero-net-investment strategy. Section 3.5 discusses the results from the previous sections, and section 3.6 contains my conclusions and suggestions for further steps.

3.2 Data

In this section, I describe various datasets for firm-level and aggregate-level stock returns. I also show how I manipulate the data for different deep learning models.

3.2.1 Aggregate-level Stock Returns

Based on results from the previous chapter, I find that larger datasets improve the performance of forecasting aggregate-level stock returns. To generate comparable results with the second chapter, I maintain the same length of market returns with additional features.

Data Descriptions For aggregate-level stock returns, I construct ten macroeconomic predictors following Vuolteenaho (2002) [91], Cochrane (2007) [22], and Goyal and Welch (2008) [37], where they provide significant in-sample predictions of aaggregate stock returns. These variables include dividend-price ratio, dividend growth rate, return on equity, earning-price ratio, book-to-market ratio, 90-day treasury bill, net equity expansion, term spread, default yield spread, and stock variance[1].

Stock return data from CRSP are based on the aggregate value-weighted stock returns, containing stocks from NYSE, AMEX, and NASDAQ. Aggregate accounting data are calculated based on the firm-level data from COMPUSTAT, aggregated based on market capitalization and deflated. The dataset contains annual market returns with features from 1950 to 2019 (same length of data as the previous chapters), and the size of dataset is 70 years by 11 features. All the variables are at the annual frequency.

Data Manipulations As discussed in the last chapter, DeepAR and DF-RNN have univariate data layouts. Each time series is imported as an individual JSON-format line, and all the data are predicted at the same time. DSSM utilizes information from stock returns only through likelihood terms, where other time series are imported as dynamic features of market returns. DeepVAR has a multivariate data layout, where all the data are imported in the same line following the JSON format. Similar to DeepAR, DeepVAR uses lags of these time series to estimate the joint conditional distribution.

Validating and Tuning Process Similar to the second chapter, I use out-of-sample R^2_{Ret}, mean absolute error (MAE), and mean absolute percentage error (MAPE) to evaluate the

[1]Dividend-price ratio and dividend growth rate are calculated directly from CRSP, value-weighted returns. Returns on equity, earning-price ratio, and book-to-market ratio are computed from COMPUSTAT, and aggregated based on market capitalization. "Net equity expansion is the ratio of 12-month moving sums of net issues by NYSE listed stocks divided by the total end-of-year market capitalization of NYSE stocks. Term spread is the difference between the long-term yield on government bonds and treasury bills. Default yield spread is the difference between BAA and AA-rated corporate bond yields. Stock variance is computed as sum of squared daily returns on the S&P 500" (Goyal and Welch (2008) [37]). Net equity expansion, term spread, default yield spread, and stock variance are computed based on the data from Amit Goyal's website.

performance. I split training and testing sets based on the common ratios, 90:10 and 85:15. Due to the time-series data format, I split the data based on the time stamps directly instead of having a random split. Also, to perform robust results, I construct two sets of training and testing datasets.

For choosing hyperparameters of models, I conduct a grid search within a selected range. For each set of hyperparameters, I fit the model on the training set and evaluate performance on the testing set with selected metrics. Controlling parameters are changed due to the different dimensionality of inputs and features. Table 3.1 presents the chosen range of hyperparameters, and other parameters are set as default.

Table 3.1: Hyper-parameters Values Range Searched based on Aggregate-level Stock Returns

	DeepAR, DSSM, DeepVAR	DF-RNN
No. of RNN layers	1,2,3,4,5,6,7	-
No. of RNN cells per layer	16,32,60,80,100	-
Batch size	16,32,48	16,32,48
Learning rate	0.1,0.01,0.001,0.0001	0.1,0.01,0.001,0.0001
No. of units per layer for global RNN	-	25,50
No. of layers for global RNN	-	2,3,4,5
No. of global factors	-	3,4,5,6,7,8
No. of units per layer for local RNN	-	16,32,48
No. of layers for local RNN	-	2,3,4,5,6

3.2.2 Firm-level Stock Returns

To further test the performance of predicting stock-level returns, I conduct an analysis of firm-level stock returns. I maintain the same frequency of stock returns with individual-level features as aggregate-level stock returns to generate comparable results.

Data Descriptions Following Gu et al. (2020) [40], I obtain annual individual equity returns and firm-level data from CRSP and COMPUSTAT for all firms listed on NYSE, AMEX, and NASDAQ. It is the hope that having a large dataset can help to improve the models' performance and avoid overfitting. Then, following Green et al. (2017) [39], I build a large dataset of firm-level predictors, including 90 annually independent factors[2]. Due to the missing values in financial statements, my firm-level sample starts in 1957 and ends in 2019, which has a shorter length than the aggregate-level sample.

Similar to the aggregate-level stock returns, I construct twelve macroeconomic predictors following Fama and French (1993) [29], Vuolteenaho (2002) [91], Cochrane (2007) [22], and Goyal and Welch (2008) [37]. These variables include dividend-price ratio, dividend growth rate, returns on equity, earning-price ratio, book-to-market ratio, 90-day treasury bill, net equity expansion, term spread, default yield spread, stock variance, SMB (small minus big), and HML (high minus low)[3]. To ensure the length of data, I use Fama-French 3 Factors (SMB & HML) instead of 5 Factors[4], which are built based on the size of firms, book-to-market ratios, and excess returns on the market.

Then, the dataset contains annual firm-level stock returns with 102 features from 1957 to 2019, and the dataset size is 355,002 observations by 102 features. The number of stocks in my sample is 30,993, with an average number of stocks per year around 5,635.

Data Manipulations Before analyzing the data, I check for missing observations in firm-level stock returns. Within the year range from 1957 to 2019, the average number of missing returns is 51, and the median number is 55, given 30,993 stocks. Using a sparse dataset can significantly weaken the performance of models, though these companies may have different listing dates. Then, I use two steps to clean the dataset. First, I try to remove stocks with

---

[2]These factors are based on Green et al. (2007) [39], and I use the SAS code from Jeremiah Green's website to retrieve these data. Then I extend the sample period and convert it from monthly to annual frequency. Following Gu et al. (2020) [40], I also adjust the data based on its frequency to avoid information leakage, which may lead to bias. Please see Section C.2 for details.

[3]SML and HML are constructed based on Fama and French (1993) [29], and they are retrieved from Kenneth French's websites.

[4]Fama-French 5 Factors are available after 1964.

missing returns in the testing and out-of-sample sets[5], as firms may delist before 2019. Then, I check the number of firms based on the different number of missing returns. I find that there are more than 10,000 firms with less than five annual returns. Thus, I remove those firms from the datasets, ending with an average of 4,652 firms per year.

Similar to the aggregate-level stock returns, DeepAR and DF-RNN have inputs containing individual stock returns without features, as they have univariate data layouts. Each time series is imported as an individual JSON-format line, and all the firm-level stock returns are predicted simultaneously.

To compare the performance based on different variables, returns, or returns with features, I keep the same firm-level stocks within the testing and out-of-sample sets. However, as DSSM and DeepVAR have multivariate inputs, I use different ways to handle missing values in the dataset and defer the discussions to Section 3.4. For DSSM, it utilizes all the information in the dataset to forecast stock returns. For each individual stock return, I assign associated lagged factors as dynamic features following the JSON format. DeepVAR has a similar format as DSSM, where all the variables are transformed into a list of lists. Different from DSSM, stock returns and predictors are predicted at the same time.

Validating and Tuning Process To generate testing results, I split training, testing, and out-of-sample sets based on the ratios, 80:10:10. Then, the training dataset is from 1957 to 2007, the testing dataset is from 2008 to 2013, and the out-of-sample dataset is from 2014 to 2019. For each year, the number of observations is different.

Then, similar to the aggregate-level predictions, I use out-of-sample R^2_{Ret}, mean absolute error (MAE), and mean absolute percentage error (MAPE) to evaluate the performance. As the dataset contains 30,993 unique firms, for both the testing and out-of-sample sets, I compute the average of firm-level evaluating metrics and form an aggregate testing statistics across firms for each model. To further evaluate the models, I sort firms into ten deciles based on book-to-market ratio, firm size, and prediction of future returns and check the

[5] Train-test splitting process is described in detail in section 3.2.2.

aggregate testing statistics[6].

Gu et al. (2020) [40] evaluate the performance of general machine learning models by maintaining the same form over time and across different stocks, which is in "contrast to standard asset pricing approaches". They use monthly data to build interactions (covariates) between stock-level and macroeconomic features to train the over-arching model recursively every year. Different from their methods, probabilistic deep learning models can utilize all the data without the loss of any information. In this case, I implement the entire dataset into various methods simultaneously, which may provide more accurate results.

For choosing hyperparameters of models, I conduct a grid search within the selected range. For each set of hyperparameters, I fit the model on the training set and examine performance on the testing and out-of-sample sets with selected metrics. Controlling parameters are changed due to the different dimensionality of inputs and features. Table 3.2 presents the chosen range of hyperparameters, and other parameters are set as default.

Table 3.2: Hyper-parameters Values Range Searched based on Firm-level Stock Returns

	DeepAR, DSSM, DeepVAR	DF-RNN
No. of RNN layers	1,2,3,4,5,6,7	-
No. of RNN cells per layer	16,32,60,80,100	-
Batch size	16,32,48	16,32,48
Learning rate	0.1,0.01,0.001,0.0001	0.01,0.001,0.0001
No. of units per layer for global RNN	-	25,50
No. of layers for global RNN	-	2,3,4,5
No. of global factors	-	3,4,5,6,7,8
No. of units per layer for local RNN	-	16,32,48
No. of layers for local RNN	-	2,3,4,5,6

[6] See Section 3.4.1 for details.

3.3 Aggregate Stock Market Returns

In Table 3.3, I show performance metrics for simple linear models with different train-test split ratios. I include up to two lags for each aggregate variable, including stock returns, dividend-price ratio, dividend growth rate, and others. For simple linear regressions, including more lagged variables weakens the performance of models.

Table 3.3: Aggregate-level Results from Simple Linear Models

	No. of Lags	MAE	MAPE	R^2
90:10 Train-Test Split	11	0.1130	0.9681	0.2358
90:10 Train-Test Split	22	0.2029	1.8187	-2.2956
85:15 Train-Test Split	11	0.1418	1.1029	-0.0738
85:15 Train-Test Split	22	0.3469	2.9659	-4.9052

This table shows statistics for out-of-sample predictions of market returns on testing dataset by using linear regressions. The second column presents the number of lagged variables for each model, including aggregate stock return, dividend-price ratio, dividend growth rate, returns on equity, earning-price ratio, book-to-market ratio, 90-day treasury bill, net equity expansion, term spread, default yield spread, and stock variance. The models are estimated based on the sample data from 1950 - 2019, CRSP and COMPUSTAT. Train-test split ratios are 90:10 and 85:15.

Table 3.4 shows performance metrics for all the probabilistic models (DeepAR, DSSM, DeepVAR, and DF-RNN, based on out-of-sample predictions). Panel A shows the results based on the first train-test dataset (90:10 ratio), while Panel B shows the results for the second train-test dataset (85:15 ratio). Given the definition of R^2, we can see that out-of-sample predictions are close to the observed data, as shown in Figure 3.1.

In Panel A, DeepVAR has the smallest MAE; DF-RNN has the highest R^2 with the second smallest MAE; and DSSM has the best MAPE with the second-largest R^2. Based on the results, we can see that DSSM and DF-RNN capture more than 70% of aggregate stock returns.

Comparing with Table 2.3 Panel D in the second chapter, we can see that with a larger

Table 3.4: Aggregate-level Results from Deep Learning Models

	No. of Features	MAE	MAPE	R^2
Panel A: 90/10 Train-Test Split				
DeepAR-90	10 features	0.1262	0.7534	0.6128
DSSM-90	10 features	0.1279	**0.4334**	0.7356
DeepVAR-90	10 features	**0.0824**	0.7084	0.5855
DF-RNN-90	10 features	0.1098	0.5632	**0.7651**
Panel B: 85/15 Train-Test Split				
DeepAR-85	10 features	0.1721	0.9603	0.4419
DSSM-85	10 features	0.2206	1.0341	0.6091
DeepVAR-85	10 features	**0.1579**	**0.5471**	**0.6314**
DF-RNN-85	10 features	0.2047	0.9916	0.5732

This table shows testing statistics for out-of-sample predictions of market returns. The second column presents the number of features for each model, including dividend-price ratio, dividend growth rate, returns on equity, earning-price ratio, book-to-market ratio, 90-day treasury bill, net equity expansion, term spread, default yield spread, and stock variance. The best results are marked in bold (R^2: the higher is better; MAE: the lower is better; MAPE: the lower is better.) The models are estimated based on the sample data from 1950 - 2019, CRSP and COMPUSTAT. Train-test split ratios are 90:10 and 85:15.

dataset, the performance of all models has improved. The best R^2 increases from 0.6580 to 0.7651; the best MAE decreases from 0.1018 to 0.0824; and the best MAPE decreases from 0.5529 to 0.4334. With more informative factors, DSSM performs better due to its automatic learning of latent dynamic systems. Further, DF-RNN significantly improves its performance by introducing the random and global factor models, which present better results than DeepAR. Figure 3.1a shows the out-of-sample predictions for the 90:10 train-test split. It shows that both DSSM and DF-RNN accurately predict the movements of future market returns, but DSSM is more precise than DF-RNN when actual market returns are small (close to zero).

Panel B presents robust results for forecasting out-of-sample market returns based on different train-test split. In this part, DeepVAR has the smallest MAE, highest R^2, and best

MAPE, and DSSM has the second-largest R^2. DSSM and DeepVAR capture more than 60% of aggregate stock returns. All the metrics are worse than the results from Table 3.4 Panel A due to the smaller training set and larger testing set. Also, R^2 is better than the result from Table 2.4 Panel D in the second chapter.

Figure 3.1b shows out-of-sample predictions for different models. Though there are a few deviations from observed market returns, most of the movements are captured by DSSM and DeepVAR. As a result, with more informative predictors, probabilistic deep learning models can make more accurate out-of-sample predictions of U.S. market returns.

3.4 Firm-level Stock Returns

In this section, I present the results from different models based on firm-level stocks. For DeepAR and DF-RNN, I only use individual stock returns without any predictors. And for DSSM and DeepVAR, due to the data layout, I use stock returns with features to forecast future annual stock returns.

To show the practical usefulness of the deep learning models' predictions, I conduct a portfolio forecasting analysis based on the pre-selected stocks. First, I sort stock into deciles based on future returns predicted by various methods. Then, I construct a zero-net-investment portfolio that longs stocks with the highest expected returns and shorts the lowest ones. For each year in the testing and out-of-sample sets, I reform value-weighted and equally-weighted portfolios based on the one-year-ahead return predictions. I find that DSSM performs the best by having over 400% cumulative excess returns (about 30% yearly excess returns).

3.4.1 Return Predictions

As mentioned in Section 3.2.2, while forecasting annual firm-level stock returns, there are many missing observations. After removing firms with less than five annual returns, I present aggregate testing statistics for simple linear regressions based on the same firm-level datasets as deep learning models in Table 3.5. For the first two rows, I only use lagged firm-level stock returns (same as DeepAR and DF-RNN) to make predictions on the testing dataset. For the third and fourth rows, I use the results from Truncated SVD and lagged returns to make predictions on the testing dataset. All the models have negative R^2. Including more lagged variables helps to reduce the MAE and MAPE, but fails to capture the volatility of individual stock returns.

Table 3.5: Firm-level Predictions Based on Simple Linear Models

	No. of Features	MAE	MAPE	R^2
Returns Only	r_{t-1}	0.1544	1.2494	-0.2861
Returns Only	r_{t-1}, r_{t-2}	0.1466	1.1619	-0.5412
Truncated SVD	3 features, r_{t-1}	0.1384	0.9245	-0.3794
Truncated SVD	3 features, r_{t-1}, r_{t-2}	0.1335	0.8437	-0.4599

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the smaller is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

DeepAR & DF-RNN

In Table 3.6, I show equally-weighted metrics for out-of-sample predictions of firm-level returns on the testing dataset. Both DeepAR and DF-RNN predict future samples based on firms' returns only. Due to the structure of global and local factors, DF-RNN trains data with both deterministic and random partitions. We can see that DF-RNN captures more volatility than DeepAR given its R^2 equals 2.33%. But, DeepAR has a smaller MAE than DF-RNN, which means that DeepAR's predictions are closer to the real observation, on average. These results are further discussed in Section 3.4.2.

Due to the time-ordered data and structure of probabilistic deep learning models, I split the testing dataset in advance (see Section 3.2.2). Though deep learning models can predict asset returns with little or no history, predictions are subject to the length of recent historical data. A large dataset with many missing observations may weaken the performance of DeepAR and DF-RNN[7]. To further test the performance, I sort the testing dataset based on assets' overall market capitalization, where middle-size firms have the smallest R^2 [8].

Table 3.6: Firm-level Predictions Based on DeepAR & DF-RNN

	MAE	MAPE	R^2
DeepAR	0.1208	0.5060	-0.0211
DF-RNN	0.1478	0.5410	0.0233

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the smaller is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Moreover, I assess forecasting performance based on pre-specified groups of stocks, which I use to build potential trading strategies. Table 3.7 shows firm-level predictions of assets' returns in the testing dataset based on different features of stocks, including market values, book-to-market ratios, and expected future cumulative returns[9]. It shows that DF-RNN forecasts returns of firms with middle sizes and neutral opportunities better than others. I further test the effectiveness of using DF-RNN to build trading strategies based on predicted returns in Section 3.4.2.

DSSM & DeepVAR

The inputs of DSSM and DeepVAR are annually firm-level stock returns with 102 features. For this dataset, I keep the same companies as the ones for DeepAR and DF-RNN. The

[7] I show the testing results on different datasets in Table C.1. The 'Filtered' dataset refers to the stocks with historical data in the training sample, and the 'Full' dataset contains all the stocks.

[8] Additional results from DeepAR and DF-RNN are shown in Section C.1, including forecasting the difference between complete and filtered datasets and top/middle/bottom stocks' performance. See Table C.1, C.2, C.3.

[9] See Table C.4 for results from DeepAR model.

Table 3.7: Firm-level Predictions Based on DF-RNN

Decile	Market Value			Book-to-Market			Model Selection		
	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Low	0.2380	0.6668	-0.0103	0.1819	0.4923	0.0041	0.1424	0.5344	0.0159
2	0.1879	0.6349	-0.0013	0.1434	0.4819	0.0055	0.1401	0.5098	0.0211
3	0.1746	0.6424	0.0084	0.1468	0.4457	0.0123	0.1364	0.5196	0.0243
4	0.1803	0.6199	0.0246	0.1385	0.4644	0.0629	0.1320	0.5451	0.0264
5	0.1436	0.5874	0.0440	0.1407	0.4989	0.0132	0.1424	0.5565	0.0228
6	0.1421	0.5272	0.0531	0.1405	0.5033	0.0129	0.1432	0.5501	0.0125
7	0.1240	0.4942	0.0511	0.1485	0.4878	0.0389	0.1389	0.5724	0.0206
8	0.1182	0.4395	0.0427	0.1574	0.5282	-0.0091	0.1509	0.5370	0.0464
9	0.0942	0.4137	0.0229	0.2012	0.6091	0.0085	0.1648	0.5436	0.0393
High	0.0747	0.3841	-0.0026	0.2063	0.5671	0.0083	0.1866	0.5417	0.0034

This table shows performance statistics for out-of-sample predictions of firm-level returns on testing dataset. Firms are split into different deciles based on market values, book-to-market ratios, or predictions of future returns. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the smaller is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

average number of missing values in the dataset is 61,204. Then, I use two different ways to deal with missing observations and build filtered dataset from the original one.

The first filtering method is based on the number of missing values and correlations among features. For each dataset, I conduct a grid-search within the chosen range of hyper-parameters (Table 3.2). The results vary a lot, and there is no specific rule about filtering data. Based on the R^2 of out-of-sample predictions on the testing set, dropping all features with missing values provides the best predictions, where R^2 equals to 3.23%. R^2 and MAE's in each model are provided in Section C.1 Table C.5.

I reduce the dimension of the dataset based on the Truncated Singular Value Decomposition (Truncated SVD). As mentioned by Reichlin et al. (2017) [74], similar to principal component analysis, Truncated SVD is good at extracting significant information from data, especially a sparse dataset. It factorizes the data matrix and truncates the smallest singular values to provide an optimal low-rank matrix. For each firm, I apply the Truncated SVD to get a matrix with reduced dimensionality. To optimize forecasting results, I choose different numbers of components for the Truncated SVD and fit DSSM and DeepVAR.

Similar to the above, Table 3.8 presents the equally-weighted metrics for out-of-sample predictions on the testing dataset. After applying the Truncated SVD, DSSM has the highest R^2, 4.52%, compared with other methods. This may be due to the architecture of DSSM and its resistance to noise. DeepVAR also shows slightly worse results than DF-RNN. But, DSSM and DeepVAR have similar MAE, which indicates on average firm-level stock returns predictions are similar. These results are further discussed in Section 3.4.2. Additional results from DSSM and DeepVAR are shown in Section C.1[10].

Table 3.8: Aggregated Firm-level Predictions Based on DSSM & DeepVAR

	MAE	MAPE	R^2
DSSM	0.1194	0.5343	0.0452
DeepVAR	0.1125	0.6051	0.0189

This table shows aggregated performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the smaller is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table 3.9 & 3.10 shows the forecasting performance within different deciles split based on market values, book-to-market ratios, or expected future returns. Similar to the table above, DSSM provides better predicting results. Predictions of middle-size firms' returns are more accurate than others, which have an R^2 of 7%. Also, expected future returns of value stocks are more precise than growth stocks.

[10] See Table C.5, C.6 for selecting number of features or dimension reductions.

Table 3.9: Firm-level Predictions Based on DSSM

Decile	Market Value			Book-to-Market			Model Selection		
	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Low	0.2058	0.9637	0.0126	0.1286	0.5090	-0.0027	0.0614	0.5008	0.1054
2	0.1447	0.5871	0.0496	0.0888	0.5006	-0.0130	0.0968	0.4902	0.0549
3	0.1359	0.5745	0.0760	0.1104	0.4630	0.0523	0.0937	0.4382	0.0118
4	0.1284	0.5267	0.0504	0.1129	0.5026	0.0622	0.1006	0.4435	0.0408
5	0.1118	0.4600	0.0602	0.1141	0.5975	0.0474	0.1085	0.4310	0.0172
6	0.1201	0.4701	0.0798	0.1197	0.5361	0.0015	0.1044	0.4776	0.0662
7	0.1069	0.4937	0.0448	0.1182	0.5457	0.0279	0.1277	0.4818	0.0584
8	0.0915	0.4533	0.0453	0.1305	0.5711	0.0524	0.1215	0.4723	0.0697
9	0.0848	0.4176	0.0114	0.1684	0.6534	0.0481	0.1516	0.4874	0.0441
High	0.0644	0.3968	0.0221	0.1990	0.5987	0.0606	0.2280	1.1203	-0.0163

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. Firms are split into different deciles based on market values, book-to-market ratios, or predictions of future returns. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the smaller is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

3.4.2 Portfolios Predictions

To further assess the performance of forecasting firm-level stock returns, I design a set of portfolios based on the sample predictions from the above models. Forecasting starts at the end of the training dataset, and I use the one-year-ahead stock return predictions to sort stocks into ten deciles. Then, I reform value-weighted and equally-weighted portfolios at the end of each year and construct a zero-net-investment portfolio that longs stocks with the highest expected returns and shorts the lowest.

Table 3.11 shows the results of models on testing samples. For each year, it shows the excess returns based on the ten-minus-one trading strategy with both value-weighted and equally weighted portfolios. Panel A shows the results from DeepAR and DF-RNN, which use firm-level stock returns only without any features. The best 10-1 strategy comes from

Table 3.10: Firm-level Predictions Based on DeepVAR

Decile	Market Value			Book-to-Market			Model Selection		
	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Low	0.2063	1.2177	0.0147	0.0910	0.5348	0.0111	0.1210	0.5956	0.0088
2	0.1467	0.7316	0.0091	0.1160	0.6516	0.0429	0.0863	0.6086	0.0097
3	0.1300	0.6759	0.0101	0.1189	0.5512	0.0125	0.1059	0.5415	0.0154
4	0.1207	0.5511	0.0287	0.1365	0.6272	0.0101	0.1038	0.5552	0.0213
5	0.1025	0.4705	0.0251	0.1024	0.5445	0.0341	0.1117	0.7488	0.0139
6	0.1093	0.4817	0.0193	0.0754	0.4923	0.0090	0.1132	0.5904	0.0138
7	0.0996	0.5729	0.0196	0.1058	0.5128	0.0401	0.1165	0.6359	0.0126
8	0.0851	0.4872	0.0050	0.1347	0.6449	-0.0011	0.1226	0.6212	0.0421
9	0.0772	0.4366	0.0147	0.1189	0.6859	0.0041	0.1595	0.7210	0.0201
High	0.0575	0.4260	0.0131	0.1353	0.8062	0.0221	0.1900	0.7111	0.0141

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. Firms are split into different deciles based on market values, book-to-market ratios, or predictions of future returns. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the smaller is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

DeepAR with an equally weighted portfolio, which, over six years, returns 513% cumulatively, and the value-weighted portfolio reports 205%. Given the metrics in Table 3.6, we can see that DF-RNN captures more movements than DeepAR, but DeepAR has a smaller MAE. Better R^2 results in more stable predictions but DeepAR may lack of precision while building strategies.

Panel B in Table 3.11 reports the results from DSSM and DeepVAR, which uses reduced-dimensional features. The best 10-1 strategy comes from DeepVAR with an equally weighted portfolio, which, over six years, returns 450% cumulatively. These results seem to be slightly better than DSSM, but DSSM provides more stable results, which can be seen in Table 3.11. Recall the performance metrics from Table 3.8, DSSM has better R^2 than DeepVAR but smaller MAE.

Table 3.12 reports the results of ten-minus-one-strategy returns on the out-of-sample

Table 3.11: Excess Returns for Pre-specified Portfolios on Testing Dataset

Panel A: DeepAR & DF-RNN

Year	DeepAR				DF-RNN			
	Value Weight		Equally Weight		Value Weight		Equally Weight	
	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret
2008	0.2740	0.1103	0.3152	0.2073	0.0596	0.0646	0.0633	0.0548
2009	0.2590	0.2706	0.2912	0.4342	0.0430	0.0919	0.0446	0.1232
2010	0.2449	0.1380	0.2726	0.2946	0.0313	0.0517	0.0321	0.0387
2011	0.2244	0.1517	0.2508	0.2280	0.0226	0.0379	0.0232	0.0185
2012	0.2245	0.1284	0.2374	0.2551	0.0163	0.0309	0.0166	0.0496
2013	0.2013	0.1042	0.2194	0.2577	0.0122	0.0465	0.0126	0.0373
Cumulative Ret	3.3875	2.0459	3.8602	5.1282	0.3194	0.7437	0.3322	0.9738

Panel B: DSSM & DeepVAR

Year	DSSM				DeepVAR			
	Value Weight		Equally Weight		Value Weight		Equally Weight	
	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret
2008	0.2828	0.1608	0.3340	0.1846	0.3972	0.1277	0.4616	0.1901
2009	0.3171	0.2448	0.3763	0.3543	0.3793	0.2002	0.4572	0.4242
2010	0.2395	0.1573	0.2871	0.2521	0.3889	0.1514	0.4361	0.2663
2011	0.2726	0.1686	0.3132	0.2212	0.3560	0.1353	0.4192	0.2294
2012	0.2910	0.1692	0.3254	0.2612	0.3609	0.1954	0.4038	0.2412
2013	0.2536	0.1404	0.2996	0.2337	0.3389	0.1149	0.3868	0.2538
Cumulative Ret	3.9836	2.5531	5.0633	4.2587	7.2752	2.1430	9.0835	4.4974

This table shows the excess returns gained for different pre-specified portfolios. Stocks are split into ten deciles based on their expected returns on the testing sample periods. "Pred Ret" and "Real Ret" provides the excess returns gained by building zero-net-investment portfolios. "Cumulative Ret" provides the cumulative excess returns over the six years. Portfolios are both value and equally weighted. The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

dataset. In Panel A, DeepAR and DF-RNN use firm-level stock returns without any features. The best 10-1 strategy comes from DeepAR with equally weighted portfolios. Compared with Panel A in Table 3.11, DeepAR shows similar cumulative returns, 499%, but annual returns have more volatility with both value-weighted and equally weighted portfolios.

Panel B shows the predictions of portfolio returns from DSSM and DeepVAR with reduced-dimensional features. Equally weighted portfolio from DSSM has the best 10-1 strategy with cumulative excess returns of 402%, which is slightly smaller than the one on the testing dataset. However, DSSM's expected and real returns are more stable than DeepAR's.

Recall the results from Table 3.6 and 3.8, we can see that DeepAR and DSSM report similar MAE, but DSSM has a much higher R^2. In this case, DSSM captures more volatility based on the information existing in firm-level stock returns.

Table 3.12: Excess Returns for Pre-specified Portfolios on Out-of-sample Dataset

Panel A: DeepAR & DF-RNN								
Year	DeepAR				DF-RNN			
	Value Weight		Equally Weight		Value Weight		Equally Weight	
	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret
2014	0.2478	0.1739	0.2910	0.3146	0.0510	0.0462	0.0501	0.0328
2015	0.2225	0.1492	0.2485	0.2551	0.0347	0.0219	0.0359	0.0368
2016	0.2103	0.1168	0.2331	0.3110	0.0262	0.0230	0.0256	0.0185
2017	0.1913	0.0994	0.2203	0.3012	0.0175	0.0161	0.0187	0.0399
2018	0.1835	0.1600	0.2057	0.2775	0.0138	-0.0069	0.0141	0.0073
2019	0.1696	0.1690	0.1900	0.3285	0.0100	0.0113	0.0105	0.0572
Cumulative Ret	2.8769	1.7986	3.3977	4.9912	0.2609	0.2022	0.2638	0.4788
Panel B: DSSM & DeepVAR								
Year	DSSM				DeepVAR			
	Value Weight		Equally Weight		Value Weight		Equally Weight	
	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret	Pred Ret	Real Ret
2014	0.3022	0.1168	0.3403	0.2250	0.4559	0.0929	0.4725	0.2146
2015	0.3366	0.1177	0.3832	0.2102	0.4322	0.0771	0.4715	0.2034
2016	0.2438	0.1716	0.2922	0.3123	0.4570	0.0814	0.4670	0.2999
2017	0.2833	0.1301	0.3235	0.2474	0.4158	0.0803	0.4637	0.2474
2018	0.2896	0.1600	0.3208	0.2271	0.4044	0.1191	0.4373	0.2475
2019	0.2716	0.1608	0.3074	0.3280	0.4178	0.1476	0.4404	0.3383
Cumulative Ret	4.4985	1.7772	5.4828	4.0158	10.3751	1.1062	11.5672	3.9920

This table shows the excess returns gained for different pre-specified portfolios. Stocks are split into ten deciles based on their expected returns on the out-of-sample periods. "Pred Ret" and "Real Ret" provides the excess returns gained by building zero-net-investment portfolios. "Cumulative Ret" provides the cumulative excess returns over the six years. Portfolios are both value and equally weighted. The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

3.5 Discussion

Based on the tables above, we can see that deep learning models significantly outperform linear models. Considering results from aggregate-level stock returns (Table 3.4), with informative features, DSSM and DF-RNN report high R^2, ranging from 73.6% to 76.5% based on a 90-10 train-test split. These results are much higher than the ones from the second chapter (65.8%) or the first chapter (35.6%). With an 85-15 train-test split, R^2 ranges from 60.9% to 63.1% for DSSM and DeepVAR. While in the second chapter, the best performance model, DF-RNN, reports an R^2 of 57.9%. Then, we can see that by including more informative features, probabilistic deep learning models perform better than before.

However, when forecasting firm-level stock returns, results from models with more than 19 features are much weaker than those with fewer but more informative features (see Table C.5). With many missing values, a sparse matrix can significantly deteriorate the predicting performance of these models (see Table C.1 and C.5). Though probabilistic deep learning models can estimate and forecast thousands of time series data at the same time, sparse matrix can still significantly affect results. Moreover, for time series data, these models are affected by the time interval between the end of training and forecasting periods, especially for time-sensitive stock returns.

For firm-level stock returns, the best performance model, DSSM, has an R^2 of 4.5%. I also split the stocks into three groups based on their market values(See Table C.2, C.3, C.7, and C.8.). In general, middle-size-firm returns can be more accurately predicted than firms with large or small market capitalization.

Moreover, I sort firms into ten deciles based on market values, book-to-market ratios, and expected returns (See Table C.4, 3.7, 3.9, and 3.10). Generally, neutral and value stocks can be more accurately predicted than growth stocks. Also, based on expected future returns, both bottom and top deciles provide roughly precise predictions.

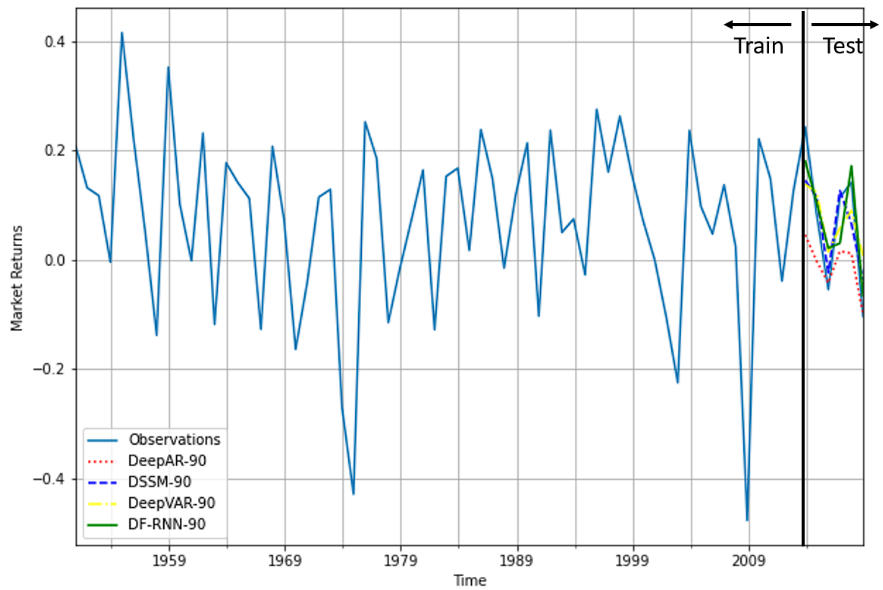
Further, I build a zero-net-investment trading strategy by longing stocks with the highest expected returns and shorting those with the lowest ones. Table 3.11 and 3.12 provide annual and cumulative returns over testing and out-of-sample periods. We can see that DSSM and DeepAR have the highest cumulative returns with equally weighted portfolios. However, DSSM provides a more stable result due to the smaller variance of both predicted and real returns. As a result, while forecasting multiple firm-level stock returns, DSSM is the optimal model with an equally weighted portfolio.

3.6 Conclusions

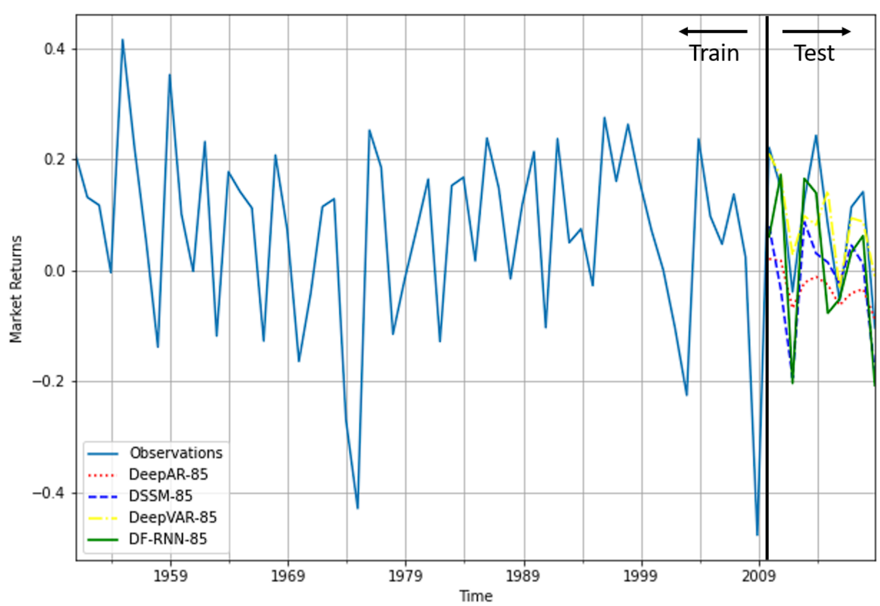
In this paper, I design more experiments to comparatively analyze probabilistic deep learning models. My findings illustrate that these models can help to improve the forecastability of stock returns in both aggregate and firm levels. By including more data, these models significantly outperform traditional vector autoregressive and linear state-space models, as shown in the first chapter. Additionally, a large dataset with an amount of missing values can deteriorate the performance of these models, which emphasizes the importance of selecting informative features. Further, estimating firm-level stock returns with features can help to build a profitable trading strategy, with a 30% average annual return(around 500% over testing or out-of-sample periods).

For predicting different time series, I find that DeepAR and DF-RNN perform the best while having a small dataset without features. But, DF-RNN provides a slightly better result and requires longer periods to train the model. Moreover, with more informative features, DSSM and DeepVAR provide more accurate results at the aggregate level. For firm-level stocks, DSSM has the optimal performance, considering its accuracy and stability. Also, based on the results from zero-net-investment trading strategies, DSSM provides the best excess returns.

Figure 3.1: Out-of-sample Predictions of Market Returns with Different Models



(a) Deep Learning with 90:10 Train-test Split



(b) Deep Learning with 85:15 Train-test Split

This plot shows two subplots that predict out-of-sample market returns based on different train-test split ratios. Part (a) shows out-of-sample predictions of market returns for different deep learning models with 90:10 train-test split. Part (b) shows out-of-sample predictions of market returns for different deep learning models with 85:15 train-test split. The models are estimated based on the sample data from 1950 - 2019, CRSP and COMPUSTAT.

BIBLIOGRAPHY

[1] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. GluonTS: Probabilistic and Neural Time Series Modeling in Python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

[2] A. Ang and G. Bekaert. Stock Return Predictability: Is it There? *The Review of Financial Studies*, 20(3):651–707, 07 2006.

[3] Susan Athey and Guido W Imbens. Machine learning methods that economists should know about. *Annual Review of Economics*, 11:685–725, 2019.

[4] A. Atkins, M. Niranjan, and E. Gerding. Financial news predicts stock market volatility better than close price. *The Journal of Finance and Data Science*, 4(2):120–137, 2018.

[5] F. Audrino, F. Sigrist, and D. Ballinari. The impact of sentiment and attention measures on stock market volatility. *International Journal of Forecasting*, 36(2):334–357, 2020.

[6] R. Ball and G. Sadka. Aggregate earnings and why they matter. *Journal of accounting literature*, 34:39–57, 2015.

[7] Robert Bartlett, Adair Morse, Richard Stanton, and Nancy Wallace. Algorithmic accountability: A legal and economic framework. Technical report, Working paper, 2020.

[8] Luca Benzoni, Pierre Collin-Dufresne, and Robert S Goldstein. Explaining asset pricing puzzles associated with the 1987 market crash. *Journal of Financial Economics*, 101(3):552–573, 2011.

[9] L. Bijl, G. Kringhaug, P. Molnár, and E. Sandvik. Google searches and stock returns. *International Review of Financial Analysis*, 45:150–156, 2016.

[10] S. B. Bonsall IV, Z. Bozanic, and P. E. Fischer. What do management earnings forecasts convey about the macroeconomy? *Journal of Accounting Research*, 51(2):225–266, 2013.

[11] M. W. Brandt and Q. Kang. On the relationship between the conditional mean and volatility of stock returns: A latent var approach. *Journal of financial economics*, 72(2):217–257, 2004.

[12] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris. Sparse and stable markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30):12267–12272, 2009.

[13] J. Y. Campbell. Stock returns and the term structure. *Journal of Financial Economics*, 18(2):373–399, 1987.

[14] J. Y. Campbell. A Variance Decomposition for Stock Returns. *The Economic Journal*, 101(405):157–179, 03 1991.

[15] J. Y. Campbell and M. Yogo. Efficient tests of stock return predictability. *Journal of Financial Economics*, 81(1):27–60, 2006.

[16] J.Y. Campbell and R.J. Shiller. The Dividend-Price Ratio and Expectations of Future Dividends and Discount Factors. *The Review of Financial Studies*, 1:195–228, 1988a.

[17] J.Y. Campbell and R.J. Shiller. Stock Prices, Earnings, and Expected Dividends. *The Journal of Finance*, 43(3):661–676, 1988b.

[18] M. Carrasco and B. Rossi. In-sample inference and forecasting in misspecified factor models. *Journal of Business & Economic Statistics*, 34(3):313–338, 2016.

[19] L. Chen, M. Pelger, and J. Zhu. Deep learning in asset pricing. *arXiv preprint arXiv:1904.00745*, 2021.

[20] J. H. Choi, A. Kalay, and G. Sadka. Earnings news, expected earnings, and aggregate stock returns. *Journal of Financial Markets*, 29:110–143, 2016.

[21] E. Chong, C. Han, and F. C Park. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 83:187–205, 2017.

[22] J. H. Cochrane. The Dog that Did not Bark: a Defense of Return Predictability. *The Review of Financial Studies*, 21(4):1533–1575, 2007.

[23] J. H. Cochrane. State-space vs. var models for stock returns. Working paper, Online, 2008.

[24] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[25] Z. Dai, X. Dong, J. Kang, and L. Hong. Forecasting stock market returns: New technical indicators and two-step economic constraint method. *The North American Journal of Economics and Finance*, 53:101216, 2020.

[26] Christine De Mol, Domenico Giannone, and Lucrezia Reichlin. Forecasting using a large number of predictors: Is bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146(2):318–328, 2008. Honoring the research contributions of Charles R. Nelson.

[27] C. H. Dow and G. C Selden. *Scientific Stock Speculation.* Windsor Books, 1920.

[28] E. F. Fama and K. R. French. Dividend yields and expected stock returns. *Journal of Financial Economics*, 22(1):3–25, 1988.

[29] Eugene F Fama and Kenneth R French. Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56, 1993.

[30] G. Feng, J. He, and N. G. Polson. Deep learning for predicting asset returns. *arXiv preprint arXiv:1804.09314*, 2018.

[31] Guanhao Feng, Stefano Giglio, and Dacheng Xiu. Taming the factor zoo: A test of new factors. *The Journal of Finance*, 75(3):1327–1370, 2020.

[32] T. Fischer and C. Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[33] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in neural information processing systems*, 30, 2017.

[34] Daniel Gedon, Niklas Wahlström, Thomas B Schön, and Lennart Ljung. Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, 54(7):481–486, 2021.

[35] Stefano Giglio and Dacheng Xiu. Asset pricing with omitted factors. *Journal of Political Economy*, 129(7):1947–1990, 2021.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[37] A. Goyal and I. Welch. A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, 21(4):1455–1508, 2008.

[38] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[39] Jeremiah Green, John RM Hand, and X Frank Zhang. The characteristics that provide independent information about average us monthly stock returns. *The Review of Financial Studies*, 30(12):4389–4436, 2017.

[40] S. Gu, B. Kelly, and D. Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

[41] S. Gu, B. Kelly, and D. Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450, 2021.

[42] J. D. Hamilton. *Time series analysis.* Princeton University Press, Princeton, NY, 1994.

[43] James B Heaton, Nick G Polson, and Jan Hendrik Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12, 2017.

[44] P. Hecht and T Vuolteenaho. Explaining returns with cash-flow proxies. *The Review of Financial Studies*, 19(1):159–194, 2006.

[45] D. Hirshleifer, K. Hou, and S. H. Teoh. Accruals,cashflows,andaggregatestockreturns. *Journal of Financial Economics*, 91:389–406, 2009.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[47] R. J. Hodrick. Dividend Yields and Expected Stock Returns: Alternative Procedures for Inference and Measurement. *The Review of Financial Studies*, 5(3):357–386, 05 1992.

[48] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[49] Z. Hu, W. Liu, J. Bian, X.e Liu, and T. Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 261–269, 2018.

[50] D. Huang, F. Jiang, J. Tu, and G. Zhou. Investor sentiment aligned: A powerful predictor of stock returns. *The Review of Financial Studies*, 28(3):791–837, 2015.

[51] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice.* OTexts, Melbourne, Australia, 3 edition, 2021.

[52] Bryan T Kelly, Seth Pruitt, and Yinan Su. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524, 2019.

[53] Chang-Jin Kim. State-space models and the kalman filter. *Unpublished*, ECON 584 Lecture Notes, 2019.

[54] S.P. Kothari, J. Lewellen, and J. B. Warner. Stock returns, aggregate earnings surprises, and behavioral finance. *Journal of Financial Economics*, 79:537–568, 2006.

[55] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[56] O. Lamont. Earnings and expected returns. *The Journal of Finance.*, 53(5):1563–1587, 1998.

[57] M. Lettau and S. Ludvigson. Consumption, aggregate wealth, and expected stock returns. *The Journal of Finance.*, 56(3):815–849, 2001.

[58] J. Lewellen. Predicting returns with financial ratios. *Journal of Financial Economics.*, 74(2):209–235, 2004.

[59] Longyuan Li, Junchi Yan, Xiaokang Yang, and Yaohui Jin. Learning interpretable deep state space model for probabilistic time series forecasting. *arXiv preprint arXiv:2102.00397*, 2021.

[60] X. Li, D. Shen, M. Xue, and W. Zhang. Daily happiness and stock returns: The case of chinese company listed in the united states. *Economic Modelling*, 64:496–501, 2017.

[61] Q. Lin. Technical analysis and stock return predictability: An aligned approach. *Journal of financial markets*, 38:103–123, 2018.

[62] R. P. Masini, M. C. Medeiros, and E. F. Mendes. Machine learning advances for time series forecasting. *Journal of economic surveys*, 2021.

[63] M. H. Miller and Franco Modigliani. Dividend policy, growth, and the valuation of shares. *The Journal of Business*, 34(4):411–433, 1961.

[64] D. D. Monache, I. Petrella, and F. Venditti. Price Dividend Ratio and Long-Run Stock Returns: A Score-Driven State Space Model. *Journal of Business and Economic Statistics*, 39(4):1054–1065, 2021.

[65] Adair Morse and Karen Pence. Technological innovation and discrimination in household finance. In *The Palgrave Handbook of Technological Finance*, pages 783–808. Springer, 2021.

[66] Sendhil Mullainathan and Jann Spiess. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.

[67] K. Nam and N. Seong. Financial news-based stock movement prediction using causality analysis of influence in the korean stock market. *Decision Support Systems*, 117:100–112, 2019.

[68] C. J. Neely, D. E. Rapach, J. Tu, and G. Zhou. Forecasting the equity risk premium: the role of technical indicators. *Management science*, 60(7):1772–1791, 2014.

[69] L. Pastor and R. F. Stambaugh. Predictive systems: Living with imperfect predictors. *The Journal of finance (New York)*, 64(4):1583–1628, 2009.

[70] P Patatoukas. Detecting news in aggregate accounting earnings: implications for stock market valuation. *Reviewe of Accounting Studies*, 19(1):134–160, 2013.

[71] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.

[72] D. E. Rapach, J. K. Strauss, and G. Zhou. International stock return predictability: What is the role of the united states? *The Journal of Finance*, 68(4):1633–1662, 2013.

[73] K. C. Rasekhschaffe and R. C Jones. Machine learning for stock selection. *Financial Analysts Journal*, 75(3):70–88, 2019.

[74] L Reichlin, C De Mol, E Gautier, D Giannone, Sendhil Mullainathan, H van Dijk, and J Wooldridge. Big data in economics: evolution or revolution? 2017.

[75] L. Reichlin, C. De Mol, E. Gautier, D. Giannone, S. Mullainathan, H. van Dijk, and J. Wooldridge. Big data in economics: evolution or revolution? In L Matyas, editor, *Economics without borders*, pages 612–632. Cambridge University Press, Cambridge, March 2017.

[76] R. Ren, D. Wu, and T. Liu. Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Systems Journal*, 13(1):760–770, 2018.

[77] A. G. Rossi. Predicting stock market returns with machine learning. *Georgetown University*, 2018.

[78] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[79] O. Rytchkov. Filtering out expected dividends and expected returns. *The quarterly journal of finance*, 2(3):1250012, 2012.

[80] G. Sadka. Understanding Stock Price Volatility: The Role of Earnings. *Journal of Accounting Research*, 45(1):199–228, 2007.

[81] G. Sadka and R. Sadka. Predictability and the earnings–returns relation. *Journal of Financial Economics*, 94(1):87–106, 2009.

[82] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32, 2019.

[83] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[84] A. A. Salisu, K. Isah, and L. O. Akanni. Improving the predictability of stock returns with bitcoin prices. *The North American Journal of Economics and Finance*, 48:857–867, 2019.

[85] Wharton School. *Wharton research data services.*, 1993.

[86] D. Stoffer. *Applied Statistical Time Series Analysis (astsa)*, 2022.

[87] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[88] J. Van Binsbergen and R. Koijen. Predictive Regressions: A Present-Value Approach. *Journal of Finance*, 65(4):1439–1471, 2010.

[89] Hal R Varian. Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2):3–28, 2014.

[90] T Vuolteenaho. Understanding the aggregate book-to-market ratio and its implications to current equity-premium expectations. Working paper, Harvard University Department of Economics, 2000.

[91] T Vuolteenaho. What drives firm-level returns? *Journal of Finance*, 57(1):233–264, 2002.

[92] Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pages 6607–6617. PMLR, 2019.

[93] M. Wolf. Stock returns and dividend yields revisited: A new way to look at an old problem. *Journal of Business and Economic Statistics*, 18(1):18–30, 2000.

[94] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, Philip S Yu, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

Appendix A

PREDICTING THE U.S. MARKET RETURNS VIA STATE-SPACE MODELS

A.1 Kalman Filter

This section shows details on the Kalman Filter for my state-space models. The discussion follows a general case and uses the state-space model with returns and dividends in Section 1.4.2 as an example. The state-space model with market returns and returns on equity in Section 1.4.3 should follow a similar setup. In this model, unobserved state variables are $\beta_t \equiv (\mu_t, g_t, \varepsilon_{\mu,t+1}, \varepsilon_{g,t+1}, \varepsilon_{d,t+1})'$, and observed variables are $y_{t+1} \equiv (r_{t+1}, \Delta d_{t+1})'$. First, we have the measurement and transition equations as follows:

$$\begin{aligned} \text{Measurement Equation: } & y_{t+1} = H\beta_t \\ \text{Transition Equation: } & \beta_t = F\beta_{t-1} + v_t \qquad v_t \overset{i.i.d}{\sim} N(0, Q) \end{aligned}$$

where H, F, and v_t follows Equations (1.14) and (1.15).

The Kalman Filter is given by [53]:

$$\beta_{0|0} = E[\beta_0], P_{0|0} = E[\beta_0\beta_0']$$

$$\beta_{t|t-1} = F\beta_{t-1|t-1}$$

$$P_{t|t-1} = FP_{t-1|t-1}F' + Q$$

$$\eta_{t|t-1} = y_t - y_{t|t-1} = y_t - H\beta_{t|t-1}$$

$$f_{t|t-1} = HP_{t|t-1}H'$$

$$\beta_{t|t} = \beta_{t|t-1} + P_{t|t-1}H'f_{t|t-1}^{-1}\eta_{t|t-1}$$

$$P_{t|t} = P_{t|t-1} - P_{t|t-1}H'f_{t|t-1}^{-1}HP_{t|t-1}$$

The likelihood is based on prediction errors ($\eta_{t|t-1}$) and the covariance matrix ($f_{t|t-1}$) through each iteration from $t = 0$ to $t = T$.

$$l(\theta) = -\frac{1}{2}\sum_t \ln\left((2\pi)^n \left|f_{t|t-1}\right|\right) - \frac{1}{2}\sum_t \eta_{t|t-1}' f_{t|t-1}^{-1}\eta_{t|t-1}$$

Then, we can maximize the conditional likelihood to estimate the vector of parameters:

$$(\phi_\mu, \phi_g, \sigma_\mu, \sigma_g, \sigma_d, \rho_{\mu g}, \rho_{\mu d}, \rho_{gd})$$

A.2 Estimation of Linearized Earning Identity

This part estimate the linearized earning identity (Equation (1.28)) by regressing the sum of the return on equity at time period t, book-to-market ratio at time period $t-1$, and negative market return and risk-free rate at time period t on the book-to-market ratio at time period t. Thus, the equation could be re-written as:

$$e_t - r_t - f_t + \theta_{t-1} \approx \alpha + \rho\theta_t + \kappa_t \tag{A.1}$$

where α and ρ are constants and estimated from the regression.

From Table A.1, results of the regression show the estimated ρ and α for the earning model. The lagged model explains 92.2% of the variance in the one-step ahead book-to-market ratio, while the discount rate is about 0.91. Though the estimation of α is not significant, the constant is eliminated through the substitution back to the identity. And the state-space-model is not affected.

Table A.1: Estimated Identity by OLS

	Estimates	S.E.	t-stats
α	0.0045	0.0179	0.2500
ρ	0.9105	0.0327	27.880
Residual SE		0.0837	
R^2		0.9220	

This table reports estimations of the regression for the above earning identity (Equation (A.1)). The dependent variable is the returns on equity minus market return and risk-free rate at time period t, then minus the lagged book-to-market ratio. The independent variable is the book-to-market ratio at time period t. Sample data from 1950 - 2019, CRSP and COMPUSTAT.

Appendix B

PROBABILISTIC DEEP-LEARNING MODELS FOR PREDICTING U.S. MARKET RETURNS

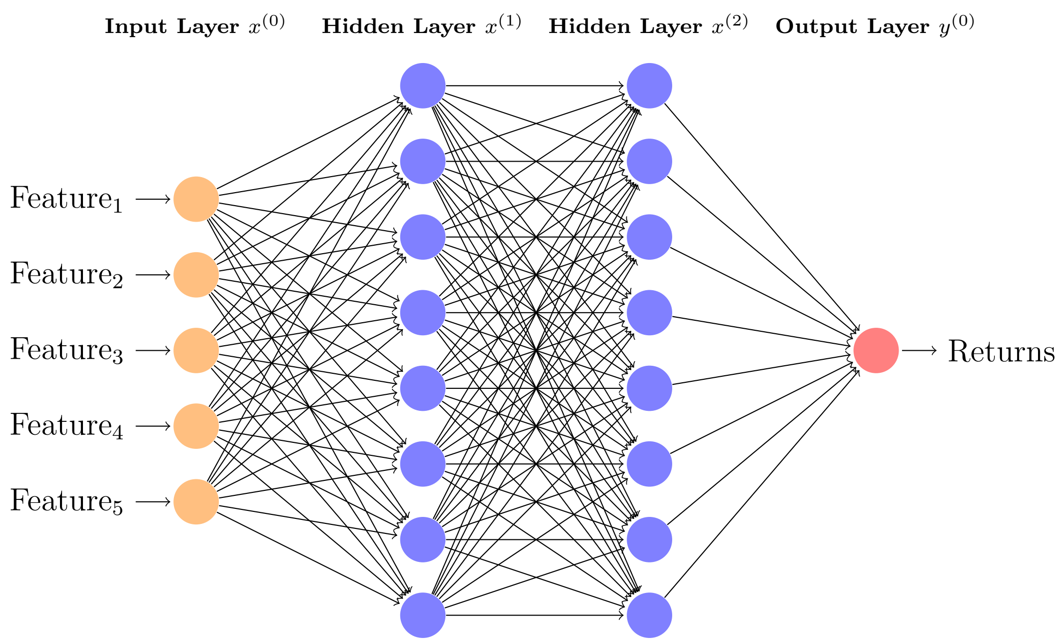
B.1 Feedforward Neural Network

Following Hornik et al. (1989) [48], Cybenko (1989) [24], and Goodfellow et al. (2016) [36], deep feedforward networks (FNN) or multilayer perceptrons (MLP) are the essential deep learning models. As shown in Figure B.1, FNNs consist of an input layer of features, one or more hidden layers, and an output layer. The input layer contains the raw features or predictor data, then the first hidden layer interacts and transforms the outputs nonlinearly from the input layer. The following layers take outputs from the previous layer to aggregate into the output layer. Similar to human brains, in each layer, there are a different number of "neurons" that analyze the transmitted information.

Feedforward neural networks compose many different functions together. For example, in Figure B.1, there are two hidden layers, and we have functions, $f^{(1)}$ and $f^{(2)}$ connected in a chain to form $f(x) = f^{(2)}(f^{(1)}(x))$. Through the training process, labels, $y \approx f^*(x)$, are approximated for each example or predictor x. But, we do not know what each individual layer does, and the learning algorithm should automatically evaluate the training data.

The number of neurons determines the dimensionality of hidden layers. Each neuron receives inputs from units of the previous layer and uses the activation function to generate its own value. There are a number of activation functions, such as rectified linear unit (ReLU), sigmoid, hyperbolic, and softmax. Then, we must choose the loss function based on the data type to form the optimization problems.

Figure B.1: Feedforward Network with 2 Hidden Layers



This figure provides a simple feedforward neural network diagram with 2 hidden layers. Orange circles denote the units in the input layer, blue circles denote the units in hidden layers, and red circle denotes the output layer. Each arrow between layers is assigned a weighting parameter. In each hidden layer, a nonlinear activation function f transforms outputs from the previous layer and passes transformed outputs to the next layer.

B.2 Training Examples

This section provides the training examples of the four models in Python. The codes below require Python packages: "json", "mxnet", "gluonts", "pandas". The data layout for Deep AR, DSSM, and DF-RNN are univariate. For Deep VAR, the data layout is multivariate. The difference in data layout is shown below in the "target" part.

B.2.1 Deep AR & DF-RNN

These are examples of training inputs for Deep AR and DF-RNN. "ListDataset" is from "gluonts" package, where each line is a time series. "start" is the start time of the time series with frequency. The start time can be different for different time series, but the frequency must be the same. The examples below have a start time on December 31st, 2008 at 12 am, with annual frequency. "target" is the target time series we want to estimate. For the examples below, the first target time series is market returns, and the second one is dividend growth rates. [1]

```
train_ds =
ListDataset(
[
{
        "start": pd.Timestamp('2008-12-31 00:00:00 UTC', freq='12M'),
        "target": [0.14955653, 0.08291643, 0.0854723, ...],
},
{
        "start": pd.Timestamp('2008-12-31 00:00:00 UTC', freq='12M'),
        "target": [0.09577301, 0.04371459, 0.08216365, ...],
},
...
],
freq = "12M"
)
```

[1] See DeepAR documentation (https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html) for more details.

B.2.2 DSSM

These are examples of training inputs for DSSM. "ListDataset" is from "gluonts" package, where each line is a time series. "start" is the start time of the time series with frequency. The start time can be different for different time series, but the frequency must be the same. The examples below have a start time on December 31st, 2008 at 12 am, with annual frequency. "target" is the target time series we want to estimate. For the examples below, the target time series is market returns. "feat_dynamic_real" is the dynamic feature associating with target time series. The first array of "feat_dynamic_real" is dividend growth rates.

```
train_ds =
ListDataset(
[
{
        "start": pd.Timestamp('2008-12-31 00:00:00 UTC', freq='12M'),
        "target": [0.14955653, 0.08291643, 0.0854723, ...],
        "feat_dynamic_real": [
        [0.09577301, 0.04371459, 0.08216365, ...],
        ...
        ]
},
]
freq = "12M"
)
```

B.2.3 DeepVAR

These are examples of training inputs for DeepVAR. "ListDataset" is from "gluonts" package, where each line is a time series. "start" is the start time of the time series with frequency. The start time can be different for different time series, but the frequency must be the same. The examples below have a start time on December 31st, 2008 at 12 am, with annual frequency. "target" is the target time series we want to estimate. For the examples below, the target time series are market returns and dividend growth rates. We use command, 'one_dim_target = False" to tell the model that this is a multi-dimensional input.

```
train_ds =
```

```
ListDataset(
[
{
        "start": pd.Timestamp('2008-12-31 00:00:00 UTC', freq='12M'),
        "target": [
        [0.14955653, 0.08291643, 0.0854723, ...],
        [0.09577301, 0.04371459, 0.08216365, ...],
        ...
        ]
},
]
freq = "12M",
one_dim_target=False
)
```

B.3 Steps on SageMaker

Probabilistic deep learning models are estimated on Amazon SageMaker. Below are the steps to run similar estimations on SageMaker. [2]

1. Create personal account.

2. Open Studio.

3. Install "gluonts", and upgrade "mxnet" and "torch".

4. Load required packages, such as: "pandas", "numpy", "json", "mxnet", "gluonts".

[2]See GluonTS tutorial (https://ts.gluon.ai/stable/tutorials/forecasting/quick_start_tutorial.html) for more details.

Appendix C

PROBABILISTIC DEEP-LEARNING MODELS FOR PREDICTING U.S. STOCK RETURNS

C.1 Additional Tables

Table C.1: Firm-level Results from DeepAR & DF-RNN with Filtered Dataset

Models	Dataset	MAE	R^2
DeepAR	Full	0.1249	-0.0226
	Filtered	0.1208	-0.0211
DF-RNN	Full	0.1344	-0.0160
	Filtered	0.1478	0.0233

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. Equally-weighted performance metrics are reported for each dataset. The best results are marked in bold (R^2: the higher is better; MAE: the lower is better.) The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table C.2: Firm-level Results from DeepAR

	Datasets	MAE	MAPE	R^2
Testing	Full	0.1208	0.5060	-0.0211
	Top 1000	0.0749	0.4123	-0.0197
	Med 1000	0.1141	0.4792	-0.0374
	Bot 1000	0.1745	0.6399	-0.0147
OOS	Full	0.0870	0.6306	-0.1436
	Top 1000	0.0519	0.4948	-0.1403
	Med 1000	0.0796	0.6448	-0.1288
	Bot 1000	0.1385	0.7560	-0.1791

This table shows performance statistics for out-of-sample predictions of firm-level returns on testing and out-of-sample datasets. Top, medium, or bottom firms are selected based on average market capitalization across sample periods. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the lower is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table C.3: Firm-level Results from DF-RNN

	Datasets	MAE	MAPE	R^2
Testing	Full	0.1478	0.5410	0.0233
	Top 1000	0.0878	0.4027	0.0111
	Med 1000	0.1435	0.5529	0.0494
	Bot 1000	0.2104	0.6480	-0.0028
OOS	Full	0.1044	0.6279	-0.1726
	Top 1000	0.0523	0.4478	-0.2220
	Med 1000	0.0971	0.6804	-0.1228
	Bot 1000	0.1723	0.7412	-0.1987

This table shows performance statistics for out-of-sample predictions of firm-level returns on testing and out-of-sample datasets. Top, medium, or bottom firms are selected based on average market capitalization across sample periods. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the lower is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table C.4: Firm-level Predictions Based on DeepAR

Decile	Market Value			Book-to-Market			Model Selection		
	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Low	0.1999	0.6981	-0.0214	0.1447	0.4912	-0.0265	0.0791	0.4652	-0.0433
2	0.1536	0.6000	-0.0083	0.1119	0.5452	-0.0720	0.0922	0.4419	-0.0155
3	0.1398	0.5259	-0.0215	0.1154	0.4347	-0.0420	0.0945	0.4439	-0.0193
4	0.1443	0.5297	-0.0420	0.1115	0.4934	-0.0024	0.1013	0.4655	-0.0092
5	0.1129	0.5195	-0.0178	0.1159	0.5220	0.0076	0.1168	0.4984	-0.0055
6	0.1129	0.4493	-0.0631	0.1149	0.5086	-0.0137	0.1173	0.4811	-0.0375
7	0.1031	0.4937	0.0144	0.1211	0.4792	-0.0069	0.1216	0.4740	0.0009
8	0.0964	0.4244	-0.0065	0.1336	0.5351	0.0123	0.1245	0.5284	-0.0243
9	0.0805	0.4290	-0.0248	0.1718	0.6131	0.0026	0.1451	0.5340	-0.0067
High	0.0647	0.3900	-0.0199	0.1774	0.5771	-0.0033	0.1716	0.6916	-0.0598

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. Firms are split into different deciles based on market values, book-to-market ratios, or predictions of future returns. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the smaller is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table C.5: Firm-level Results from DSSM & DeepVAR with Reduced Dimensions

Models	No. of Features	MAE	R^2
DSSM	89	0.1556	0.0268
	50	0.1439	0.0286
	43	0.1458	0.0218
	27	0.1458	0.0290
	19	**0.1336**	**0.0323**
DeepVAR	89	0.1470	-0.0896
	50	0.1392	-0.0878
	43	0.1331	-0.0624
	27	**0.1330**	-0.0715
	19	0.1333	**-0.0396**

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. For each dataset, I choose different number of features based on filling rates and correlations among these variables. Equally-weighted performance metrics are reported for each dataset. The best results are marked in bold (R^2: the higher is better; MAE: the lower is better.) The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table C.6: Firm-level Results from DSSM & DeepVAR with Truncated SVD

Models	No. of Components	MAE	R^2
DSSM	2	0.1227	0.0233
	3	**0.1194**	**0.0452**
	4	0.1233	0.02496
	5	0.1219	0.03034
DeepVAR	2	0.1130	-0.0213
	3	**0.1125**	**0.0189**
	4	0.1133	-0.0286
	5	0.1142	-0.0282

This table shows performance statistics for out-of-sample predictions of firm-level returns on the testing dataset. For each dataset, I choose a different number of components for the truncated SVD method and reduce the dimensions of datasets. Equally-weighted performance metrics are reported for each dataset. The best results are marked in bold (R^2: the higher is better; MAE: the lower is better.) The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table C.7: Firm-level Results from DSSM

	Datasets	MAE	MAPE	R^2
Testing	Full	0.1194	0.5343	0.0452
	Top 1000	0.0935	0.4463	0.0407
	Med 1000	0.1179	0.4995	0.0592
	Bot 1000	0.1453	0.6224	0.0498
OOS	Full	0.0796	0.5899	-0.2972
	Top 1000	0.0538	0.4593	-0.2966
	Med 1000	0.0724	0.5715	-0.3099
	Bot 1000	0.1052	0.7150	-0.3032

This table shows performance statistics for out-of-sample predictions of firm-level returns on testing and out-of-sample datasets. All the datasets have reduced dimensions by applying Truncated SVD. Top, medium, or bottom firms are selected based on average market capitalization across sample periods. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the lower is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

Table C.8: Firm-level Results from DeepVAR

	Datasets	MAE	MAPE	R^2
Testing	Full	0.1125	0.6051	0.0189
	Top 1000	0.0857	0.4809	0.0144
	Med 1000	0.0110	0.5401	0.0163
	Bot 1000	0.1412	0.7294	0.0096
OOS	Full	0.1079	0.9659	-0.1948
	Top 1000	0.0816	0.7891	-0.2039
	Med 1000	0.1037	0.9701	-0.1925
	Bot 1000	0.1352	1.1400	-0.1816

This table shows performance statistics for out-of-sample predictions of firm-level returns on testing and out-of-sample datasets. All the datasets have reduced dimensions by applying Truncated SVD. Top, medium, or bottom firms are selected based on average market capitalization across sample periods. Equally-weighted performance metrics are reported for each dataset(R^2: the higher is better; MAE: the lower is better; MAPE: the lower is better). The models are estimated based on the sample data from 1957 - 2019, CRSP and COMPUSTAT. Train-test-oos split ratios are 80:10:10.

C.2 Firm-level Variable Descriptions

This section includes tables from Green et al. (2007) [39], which describe the firm-level variables. I drop four variables, *chmom*, *maxret*, *mom1m*, and *mom6m*, from the dataset, as they are monthly momentum or returns, or daily variables.

Acronym	Paper's author(s)	Date, Journal	Definition of the characteristic-based anomaly variable
absacc	Bandyopadhyay, Huang & Wirjanto	2010, WP	Absolute value of *acc*.
acc	Sloan	1996, TAR	Annual income before extraordinary items (*ib*) minus operating cash flows (*oancf*) divided by average total assets (*at*); if *oancf* is missing then set to change in *act* - change in *che* - change in *lct* + change in *dlc* + change in *txp-dp*.
aeavol	Lerman, Livnat & Mendenhall	2007, WP	Average daily trading volume (*vol*) for 3 days around earnings announcement minus average daily volume for 1-month ending 2 weeks before earnings announcement divided by 1-month average daily volume. Earnings announcement day from Compustat quarterly (*rdq*).
age	Jiang, Lee & Zhang	2005, RAS	Number of years since first Compustat coverage.
agr	Cooper, Gulen & Schill	2008, JF	Annual percent change in total assets (*at*).
baspread	Amihud & Mendelson	1989, JF	Monthly average of daily bid-ask spread divided by average of daily spread.
beta	Fama & MacBeth	1973, JPE	Estimated market beta from weekly returns and equal weighted market returns for 3 years ending month *t*-1 with at least 52 weeks of returns.
betasq	Fama & MacBeth	1973, JPE	Market beta squared.
bm	Rosenberg, Reid & Lanstein	1985, JPM	Book value of equity (*ceq*) divided by end of fiscal-year-end market capitalization.
bm_ia	Asness, Porter & Stevens	2000, WP	Industry adjusted book-to-market ratio.
cash	Palazzo	2012, JFE	Cash and cash equivalents divided by average total assets.
cashdebt	Ou & Penman	1989, JAE	Earnings before depreciation and extraordinary items (*ib*+*dp*) divided by avg. total liabilities (*lt*).
cashpr	Chandrashekar & Rao	2009, WP	Fiscal year end market capitalization plus long term debt (*dltt*) minus total assets (*at*) divided by cash and equivalents (*che*).
cfp	Desai, Rajgopal & Venkatachalam	2004, TAR	Operating cash flows divided by fiscal-year-end market capitalization.
cfp_ia	Asness, Porter & Stevens	2000, WP	Industry adjusted *cfp*.
chatoia	Soliman	2008, TAR	2-digit SIC - fiscal-year mean adjusted change in sales (*sale*) divided by average total assets (*at*).
chcsho	Pontiff & Woodgate	2008, JF	Annual percent change in shares outstanding (*csho*).
chempia	Asness, Porter & Stevens	1994, WP	Industry-adjusted change in number of employees.
chfeps	Hawkins, Chamberlin & Daniel	1984, FAJ	Mean analyst forecast in month prior to fiscal period end date from I/B/E/S summary file minus same mean forecast for prior fiscal period using annual earnings forecasts.
chinv	Thomas & Zhang	2002, RAS	Change in inventory (*inv*) scaled by average total assets (*at*).
chmom	Gettleman & Marks	2006, WP	Cumulative returns from months *t*-6 to *t*-1 minus months *t*-12 to *t*-7.
chnanalyst	Scherbina	2007, WP	Change in *nanalyst* from month *t*-3 to month *t*.
chpmia	Soliman	2008, TAR	2-digit SIC - fiscal-year mean adjusted change in income before extraordinary items (*ib*) divided by sales (*sale*).
chtx	Thomas & Zhang	2011, JAR	Percent change in total taxes (*txtq*) from quarter *t*-4 to *t*.

cinvest	Titman, Wei & Xie	2004, JFQA	Change over one quarter in net PP&E (*ppentq*) divided by sales (*saleq*) - average of this variable for prior 3 quarters; if *saleq* = 0, then scale by 0.01.
convind	Valta	2016, JFQA	An indicator equal to 1 if company has convertible debt obligations.
currat	Ou & Penman	1989, JAE	Current assets / current liabilities.
depr	Holthausen & Larcker	1992, JAE	Depreciation divided by PP&E.
disp	Diether, Malloy & Scherbina	2002, JF	Standard deviation of analyst forecasts in month prior to fiscal period end date divided by the absolute value of the mean forecast; if *meanest* = 0, then scalar set to 1. Forecast data from I/B/E/S summary files.
divi	Michaely, Thaler & Womack	1995, JF	An indicator variable equal to 1 if company pays dividends but did not in prior year.
divo	Michaely, Thaler & Womack	1995, JF	An indicator variable equal to 1 if company does not pay dividend but did in prior year.
dolvol	Chordia, Subrahmanyam & Anshuman	2001, JFE	Natural log of trading volume times price per share from month *t*-2.
dy	Litzenberger & Ramaswamy	1982, JF	Total dividends (*dvt*) divided by market capitalization at fiscal year-end.
ear	Kishore, Brandt, Santa-Clara & Venkatachalam	2008, WP	Sum of daily returns in three days around earnings announcement. Earnings announcement from Compustat quarterly file (*rdq*).
egr	Richardson, Sloan, Soliman & Tuna	2005, JAE	Annual percent change in book value of equity (*ceq*).
ep	Basu	1977, JF	Annual income before extraordinary items (*ib*) divided by end of fiscal year market cap.
fgr5yr	Bauman & Dowen	1988, FAJ	Most recently available analyst forecasted 5-year growth.
gma	Novy-Marx	2013, JFE	Revenues (*revt*) minus cost of goods sold (*cogs*) divided by lagged total assets (at).
grCAPX	Anderson & Garcia-Feijoo	2006, JF	Percent change in capital expenditures from year *t*-2 to year *t*.
grltnoa	Fairfield, Whisenant & Yohn	2003, TAR	Growth in long term net operating assets.
herf	Hou & Robinson	2006, JF	2-digit SIC - fiscal-year sales concentration (sum of squared percent of sales in industry for each company).
hire	Bazdresch, Belo & Lin	2014, JPE	Percent change in number of employees (*emp*).
idiovol	Ali, Hwang & Trombley	2003, JFE	Standard deviation of residuals of weekly returns on weekly equal weighted market returns for 3 years prior to month end.
ill	Amihud	2002, JFM	Average of daily (absolute return / dollar volume).
indmom	Moskowitz & Grinblatt	1999, JF	Equal weighted average industry 12-month returns.
invest	Chen & Zhang	2010, JF	Annual change in gross property, plant, and equipment (*ppegt*) + annual change in inventories (*invt*) all scaled by lagged total assets (*at*).
IPO	Loughran, Ritter & Ritter	1995, JF	An indicator variable equal to 1 if first year available on CRSP monthly stock file.
lev	Bhandari	1988, JF	Total liabilities (lt) divided by fiscal year end market capitalization.
lgr	Richardson, Sloan, Soliman & Tuna	2005, JAE	Annual percent change in total liabilities (*lt*).
maxret	Bali, Cakici & Whitelaw	2011, JFE	Maximum daily return from returns during calendar month *t*-1.
mom12m	Jegadeesh	1990, JF	11-month cumulative returns ending one month before month end.
mom1m	Jegadeesh & Titman	1993, JF	1-month cumulative return.
mom36m	Jegadeesh & Titman	1993, JF	Cumulative returns from months *t*-36 to *t*-13.

mom6m	Jegadeesh & Titman	1993, JF	5-month cumulative returns ending one month before month end.
ms	Mohanram	2005, RAS	Sum of 8 indicator variables for fundamental performance.
mve	Banz	1981, JFE	Natural log of market capitalization at end of month *t*-1.
mve_ia	Asness, Porter & Stevens	2000, WP	2-digit SIC industry-adjusted fiscal year-end market capitalization.
nanalyst	Elgers, Lo & Pfeiffer	2001, TAR	Number of analyst forecasts from most recently available I/B/E/S summary files in month prior to month of portfolio formation. *nanalyst* set to zero if not covered in I/B/E/S summary file.
nincr	Barth, Elliott & Finn	1999, JAR	Number of consecutive quarters (up to eight quarters) with an increase in earnings (*ibq*) over same quarter in the prior year.
operprof	Fama & French	2015, JFE	Revenue minus cost of goods sold - SG&A expense - interest expense divided by lagged common shareholders' equity.
orgcap	Eisfeldt & Papanikolaou	2013, JF	Capitalized SG&A expenses.
pchcapx_ia	Abarbanell & Bushee	1998, TAR	2-digit SIC - fiscal-year mean adjusted percent change in capital expenditures (*capx*).
pchcurrat	Ou & Penman	1989, JAE	Percent change in *currat*.
pchdepr	Holthausen & Larcker	1992, JAE	Percent change in *depr*.
pchgm_pchsale	Abarbanell & Bushee	1998, TAR	Percent change in gross margin (*sale-cogs*) minus percent change in sales (*sale*).
pchquick	Ou & Penman	1989, JAE	Percent change in *quick*.
pchsale_pchinvt	Abarbanell & Bushee	1998, TAR	Annual percent change in sales (*sale*) minus annual percent change in inventory (*invt*).
pchsale_pchrect	Abarbanell & Bushee	1998, TAR	Annual percent change in sales (*sale*) minus annual percent change in receivables (*rect*).
pchsale_pchxsga	Abarbanell & Bushee	1998, TAR	Annual percent change in sales (*sale*) minus annual percent change in SG&A (*xsga*).
pchsaleinv	Ou & Penman	1989, JAE	Percent change in *saleinv*.
pctacc	Hafzalla, Lundholm & Van Winkle	2011, TAR	Same as *acc* except that the numerator is divided by the absolute value of *ib*; if *ib* = 0 then *ib* set to 0.01 for denominator.
pricedelay	Hou & Moskowitz	2005, RFS	The proportion of variation in weekly returns for 36 months ending in month *t* explained by 4 lags of weekly market returns incremental to contemporaneous market return.
ps	Piotroski	2000, JAR	Sum of 9 indicator variables to form fundamental health score.
quick	Ou & Penman	1989, JAE	(current assets - inventory) / current liabilities.
rd	Eberhart, Maxwell & Siddique	2004, JF	An indicator variable equal to 1 if R&D expense as a percentage of total assets has an increase greater than 5%.
rd_mve	Guo, Lev & Shi	2006, JBFA	R&D expense divided by end-of-fiscal-year market capitalization.
rd_sale	Guo, Lev & Shi	2006, JBFA	R&D expense divided by sales (*xrd/sale*).
realestate	Tuzel	2010, RFS	Buildings and capitalized leases divided by gross PP&E.
retvol	Ang, Hodrick, Xing & Zhang	2006, JF	Standard deviation of daily returns from month *t*-1.
roaq	Balakrishnan, Bartov & Faurel	2010, JAE	Income before extraordinary items (*ibq*) divided by one quarter lagged total assets (*atq*).

roavol	Francis, LaFond, Olsson & Schipper	2004, TAR	Standard deviaiton for 16 quarters of income before extraordinary items (*ibq*) divided by average total assets (*atq*).
roeq	Hou, Xue & Zhang	2014, RFS	Earnings before extraordinary items divided by lagged common shareholders' equity.
roic	Brown & Rowe	2007, WP	Annual earnings before interest and taxes (*ebit*) minus non-operating income (*nopi*) divided by non-cash enterprise value (*ceq+lt-che*).
rsup	Kama	2009, JBFA	Sales from quarter t minus sales from quarter *t*-4 (*saleq*) divided by fiscal-quarter-end market capitalization (*cshoq* * *prccq*).
salecash	Ou & Penman	1989, JAE	Annual sales divided by cash and cash equivalents.
saleinv	Ou & Penman	1989, JAE	Annual sales divided by total inventory.
salerec	Ou & Penman	1989, JAE	Annual sales divided by accounts receivable.
secured	Valta	2016, JFQA	Total liability scaled secured debt.
securedind	Valta	2016, JFQA	An indicator equal to 1 if company has secured debt obligations.
sfe	Elgers, Lo & Pfeiffer	2001, TAR	Analysts mean annual earnings forecast for nearest upcoming fiscal year from most recent month available prior to month of portfolio formation from I/B/E/S summary files scaled by price per share at fiscal quarter end.
sgr	Lakonishok, Shleifer & Vishny	1994, JF	Annual percent change in sales (*sale*).
sin	Hong & Kacperczyk	2009, JFE	An indicator variable equal to 1 if a company's primary industry classification is in smoke or tobacco, beer or alcohol, or gaming.
SP	Barbee, Mukherji, & Raines	1996, FAJ	Annual revenue (*sale*) divided by fiscal-year-end market capitalization.
std_dolvol	Chordia, Subrahmanyam & Anshuman	2001, JFE	Monthly standard deviation of daily dollar trading volume.
std_turn	Chordia, Subrahmanyam, &Anshuman	2001, JFE	Monthly standard deviation of daily share turnover.
stdacc	Bandyopadhyay, Huang & Wirjanto	2010, WP	Standard deviation for 16 quarters of accruals (*acc* measured with quarterly Compustat) scaled by sales; if *saleq* = 0, then scale by 0.01.
stdcf	Huang	2009, JEF	Standard deviation for 16 quarters of cash flows divided by sales (*saleq*); if *saleq* = 0, then scale by 0.01. Cash flows defined as *ibq* minus quarterly accruals.
sue	Rendelman, Jones & Latane	1982, JFE	Unexpected quarterly earnings divided by fiscal-quarter-end market cap. Unexpected earnings is I/B/E/S actual earnings minus median forecasted earnings if available, else it is the seasonally differenced quarterly earnings before extraordinary items from Compustat quarterly file.
tang	Almeida & Campello	2007, RFS	Cash holdings + 0.715 × receivables + 0.547 × inventory + 0.535 × PPE/ total assets.
tb	Lev & Nissim	2004, TAR	Tax income, calculated from current tax expense divided by maximum federal tax rate, divided by income before extraordinary items.
turn	Datar, Naik & Radcliffe	1998, JFM	Average monthly trading volume for most recent 3 months scaled by number of shares outstanding in current month.
zerotrade	Liu	2006, JFE	Turnover weighted number of zero trading days for most recent 1 month.