Developing Data-Driven Methods Using Machine Learning in Operations and Finance

by

Davood Pirayesh Neghab

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Industrial Engineering and Operations Management

June 10, 2021

Developing Data-Driven Methods Using Machine Learning in Operations and Finance

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Davood Pirayesh Neghab

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Fikri Karaesmen (Advisor)

_____

Asst. Prof. Pelin Canbolat

_____

Assoc. Prof. Uğur Çelikyurt

_____

Assoc. Prof. Reza Bradrania

_____

Asst. Prof. Ethem Çanakoğlu

Date: _____

ABSTRACT

With the recent advances in data collection and analysis technologies, data-driven approaches to operations management and financial problems have gained traction. In particular, machine learning methods are increasingly being integrated into optimization problems. This integration facilitates translating the historical data into prescriptive solutions in comprehensive frameworks. This thesis aims at building data-driven methods to link the data with the actions and improve the goals in operational and financial decision systems. We consider an integrated learning and optimization approach using machine learning techniques for optimizing strategy for decision-makers facing a complex problem with additional information about the state of the system. We give algorithms based on integrating optimization, neural networks, and adapting time series properties, and develop models that are capable of estimating nonlinear relations between data. We focus on three problems from the fields of inventory, financial investment, and firms evaluation.

In the context of inventory, we consider an integrated learning and optimization problem for optimizing a newsvendor's strategy facing a complex correlated demand with additional information about the unobservable state of the system. We, therefore, combine estimation, inference, and optimization using a multi-layered neural network. To assess the performance of this integrated approach, we compare the results from our approach against data-based methods that ignore the hidden factor information or that employ separate inference and optimization steps. Numerical examples on both a synthetic data set and real data which might have an unobservable state demonstrate that our approach compares favorably against the other benchmarks.

In the context of financial investment, we propose a new approach to asset al-

location based on machine learning; it analyzes historical market states and asset returns and identifies the optimal portfolio choice in a new period when new observations become available. In this approach, we directly relate state variables to portfolio weights, rather than first modeling the return distribution and subsequently estimating the portfolio choice. The method captures nonlinearity among the state (predicting) variables and portfolio weights without assuming any particular distribution of returns and other data, without fitting a model with a fixed number of predicting variables to data, and without estimating any parameters. The empirical results for a portfolio of stock and bond indices show the proposed approach generates a more efficient outcome compared to traditional methods and is robust in using different objective functions across different sample periods.

In the last project, we propose a nonlinear approach based on stochastic frontier analysis and machine learning to estimate the pricing efficiency and the level of premarket inefficiencies for initial public offerings (IPOs). This approach enables us to estimate IPO pricing efficiency using information available before the IPO day and without any distributional assumptions for deliberate underpricing in the premarket that have been documented in the literature. We apply the proposed approach in the U.S. IPO market and show that only a few determinants of the value of firms impact the pricing and underpricing of IPOs.

ACKNOWLEDGMENTS

Foremost, I would like to express my deep and sincere gratitude to my advisor, Prof. Fikri Karaesmen for his continuous support. Without his expertise and persistent help, this dissertation would not have been possible. His timely advice, prompt inspiration, and scientific approach have helped me to a very great extent to accomplish this task.

In addition, I owe a deep sense of gratitude to my primary research supervisor Asst. Prof. Gabor Rudolf, who is unfortunately no longer with us, for giving me the opportunity to research hot topics such as data-driven methods and machine learning and providing invaluable guidance through this research. His vision, sincerity, immense knowledge, and motivation have deeply inspired me. It was a great privilege and honor to work with him.

My sincere thanks go to Assoc. Prof. Reza Bradrania, Department of Business at the University of South Australia, for providing an opportunity to be a visiting researcher at that university and trusting me so that I could apply and extend my thesis in Finance. He convincingly conveyed a spirit of adventure about research.

I also wish to thank other committee members, Asst. Prof. Pelin Canbolat, Assoc. Prof. Uğur Çelikyurt, and Asst. Prof. Ethem Çanakoğlu, for their valuable advice and fruitful discussions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

HMM	Hidden Markov Model
ML	Machine Learning
DNN	Deep Neural Network
ANN	Artificial Neural Network
ERM	Empirical Risk Minimization
SFA	Stochastic Frontier Analysis
SGD	Stochastic Gradient Descent
BPA	Backpropagation Algorithm
IPO	Initial Public Offering
COLS	Corrected Ordinary Least Square

Chapter 1

INTRODUCTION

In this chapter, we provide the research overview and the sub-contexts of this thesis. We then briefly explain the problems and their motivation that are addressed in the next chapters.

1.1 Research Overview

There is strong evidence that business performance can be improved via data-driven decision-making, big data technologies, and data-science techniques (Provost and Fawcett, 2013). The connection between big data as an important aspect of computational science with several areas of operations, finance, marketing, management, and psychology, and research issues that are related to the various disciplines has increasingly gained attention (Chang et al., 2018). On one hand, enormous recent advances in new technologies such as artificial intelligence and machine learning techniques as well as data-driven approaches to solving problems have paved the way for cheaply gathering real-time data and boiling down the complex decision-making problems to optimization under constraints based on data (Quinn et al., 2014; Wang et al., 2018; Cioffi et al., 2020). On the other hand, data itself can be beneficial for shortening decision steps, increasing financial profitability, and easing the planning of smart investments, as well as providing timely feedback for continuous business model re-engineering, reducing the distance between forecasts and actual occurrences (Moro Visconti and Morea, 2019).

With the complex nature of the recently emerged field of data-driven decision-making approaches, the classical optimization and statistical models seem inade-

quate. Most Operations Research/Management Science models focus on optimizing sub-systems with limited scope by attempting to simplify the real problem or concentrating on a small part of a whole complex system (Besiou and Van Wassenhove, 2015). Despite various approaches in OR/MS with optimization methods, none of them seems to be perfectly applicable for real-world challenges to consider complex systems and their states (Sodhi and Tang, 2008). The interest in data-driven approaches has increased lately, mainly due to the growing complexity within the dynamics of the systems. This thesis is motivated by the need of developing implementable data-driven decision systems and devising innovative solutions to real-world complex problems in operations, business, finance, and management. We aim at answering the following research questions: *what is the data-driven solution to combine time series models with machine learning techniques? how to design methods that overcome the problem of paying equal attention to costly and cheap mistakes, when the optimizer is blind to the confidence of the learner in its estimates for different regions of the problem? how do machine learning techniques provide interpretational insight into complex models?*

We pursue our goals in both of data-driven methods, which allow integrating estimation and optimization steps, and machine learning techniques. We propose to combine these two streams to provide reliable system modeling that are capable of considering complex relations in systems where classical methods fail. In many problems of operations and finance, the decision maker has to decide under uncertainty. Recent advances in data collection and new technologies facilitate translating historical data into the decisions. These data, also referred to as *feature, predictor*, or *explanatory*, are some information that is gathered and recorded besides the main variables. A belief in the existence of an explanatory relationship between data resulted in the new approach of *Data-driven* methods. Figure 1.1 shows the main approach and two types of solution methods; first, those estimate parameters and then decide based on estimations and called *predictors*. Second, those make decisions directly from data and called *prescriptors*.

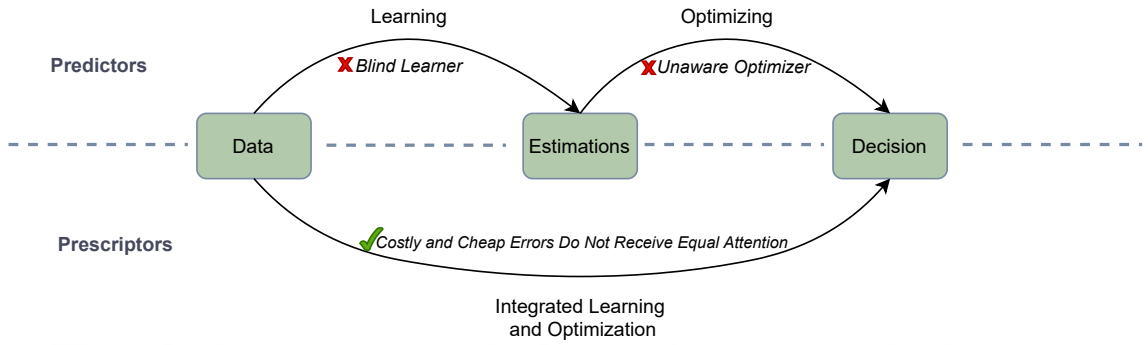


Figure 1.1: Data-driven methods.

The first method has some drawbacks that make it less applicable. In the learning step of this method, the learner is blind to the main problem and objective of the optimization, while the optimizer is also unaware of the learner's confidence and what is happening in the learning process. Due to these setbacks, the second approach that integrates learning and optimization has emerged to solve these issues efficiently. In this method, costly and cheap errors do not receive equal attention and the estimation errors are alleviated. Benefiting from non-parametric models and making the decisions more dynamically are other advantages of this approach. Furthermore, unlike parametric and classic approaches, the estimation and optimization steps are integrated into this method. This approach has been proved beneficial and has been extended to other problems in operations and finance. (Liyanage and Shanthikumar, 2005; Huh et al., 2011; Sachs, 2015; He et al., 2012; Ban and Rudin, 2019; Bertsimas and Kallus, 2014; Aït-Sahalia and Brandt, 2001; Brandt et al., 2009; Malandri et al., 2018).

In this thesis, we provide comprehensive decision-making frameworks by developing data-driven approaches and combining ideas from machine learning and optimization in operations and finance. Figure 1.2 shows different areas and fields of the present study. In the first two problems of inventory and asset allocation, where the uncertainty characterizes the nature of these problems, we apply the new data-driven method and compare the results with those obtained by traditional methods. We

then, in the second part of the asset allocation, and firm evaluation, focus on interpreting the models provided by machine learning. Machine learning tools are usually considered as a black box that is difficult to be analyzed. Therefore, we use several analytical methods to shed light on the final models and provide useful insights and advice for decision-makers who use new methods.

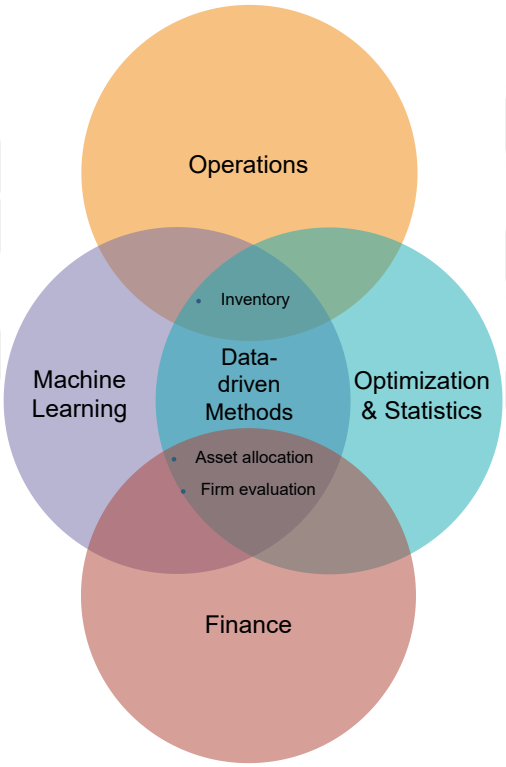


Figure 1.2: Different areas of the study.

1.2 Data-Driven Inventory

The first project relates to inventory systems. In particular, we consider a single-period inventory problem with random demand with both directly observable and unobservable features that impact the demand distribution. The single-period random demand inventory problem is one of the central problems in inventory control. In the standard version of the problem, it is assumed that the inventory manager chooses an order quantity before observing the random demand. The mismatch between the order

quantity may lead to unsatisfied demand or unsold items and the implied costs of a unit lost demand and an unsold item are not symmetrical. The simplifying assumption on random demand is that its probability distribution is known with certainty and this distribution is independent and identically distributed in each period. Recent interest in data-driven approaches to inventory management has stimulated renewed interest in general versions of this problem where the random demand depends on multiple factors that are observed before ordering. In addition, there is past data on the observed factors and the corresponding demand that was realized that can guide a data-dependent ordering decision.

In this project, we contribute to the above stream of literature by investigating a single-period random demand inventory problem where random demand in each period is dependent on an unobservable factor, which undergoes a Markovian model, in addition to some observable factors. We assume that past data is available on the observable factors and the corresponding demand that was observed. On the other hand, information about the unobservable factor has to be inferred from past data. The traditional approach to such a problem would be to first estimate the underlying model parameters for the effect of the observable factors on demand and infer the hidden state information and its effects. Once estimation and inference take place, the optimization problem would be solved in the second step. However, some of the recent research on this problem (see Ban and Rudin (2019) for example) has demonstrated the advantages of integrating the estimation step of the parameters for the observable factors with the optimization step using tools from machine learning. We pursue this integrated estimation and optimization approach in a model which has the additional complexity of an unobservable factor. We, therefore, combine estimation, inference, and optimization using a multi-layered neural network. To assess the performance of this integrated approach, we compare the results from our approach against data-based methods that ignore the hidden factor information or that employ separate inference and optimization steps. Numerical examples on both a synthetic data set and real data which might have an unobservable state demonstrate

that our approach compares favorably against the other benchmarks.

1.3 State-Dependent Asset Allocation

An investor faces a multi-stage problem when she constructs a portfolio of assets. Portfolio construction includes several related elements such as setting investment strategy and objectives, the decision on suitable portions of portfolio assets, known as asset allocation, security selection, and portfolio monitoring and rebalancing according to portfolio strategies. In this thesis, we focus on asset allocation, as a classical problem in the portfolio and fund management. This problem includes portfolio composition by deciding on optimal weights associated with each asset class to achieve the investment goal, characterized by risk and return. Modern portfolio theory based on the mean-variance framework is valid for the portfolio choice of an investor if returns are normally distributed (Markowitz, 1952; Sharpe, 1966). However, it is well documented that returns do not follow normal distributions. (Ekholm and Pasternack, 2005; Leland, 1999; Kat and Brooks, 2001; Agarwal and Naik, 2004; Malkiel and Saha, 2005). When the normality of returns distribution is violated, the mean-variance model and the Sharpe ratio lead to an inefficient estimation of risk, and the Sharpe ratio results in a misleading performance evaluation of assets. This imperfection is due to ignoring higher moments of returns distribution. To overcome this well-known limitation of the Sharpe ratio, several alternative performance measures such as Gini (Shalit and Yitzhaki, 1984), Mean Absolute Deviation (MAD) (Konno and Yamazaki, 1991), Minimax (Young, 1998), Rachev (Biglova et al., 2004), and Conditional Value at Risk (CVaR) (Martin et al., 2003) are suggested. Cogneau and Hübner (2009) categorize these measures in addition to some others as standardized risk-adjusted performance measures. In particular, we focus on miscellaneous types of relative measures that are computed as a ratio dividing the performance by a risk measure. The sub-classes of this group of performance measures are made according to how risk is measured. We choose a set of performance measures that include the varieties in risk definition proposed in the literature.

In this thesis, we propose a new approach to asset allocation that is based on machine learning; it analyzes historical market states and asset returns and identifies the optimal portfolio choice in a new period when new observations become available. In this approach, we directly relate state variables to portfolio weights, rather than firstly modeling the return distribution and subsequently estimating the portfolio choice. The method captures nonlinearity among the state (predicting) variables and portfolio weights without assuming any particular distribution of returns and other data, without fitting a model with a fixed number of predicting variables to data, and without estimating any parameters. In this approach, instead of first modeling moments and various features of the conditional return distribution, and subsequently characterizing optimal trading strategies, we skip the estimation of the conditional return distribution and directly decide on portfolio weights using predicting variables. This is important because the relationship between the portfolio weights – as the complex functions of the return distribution – and the predictors is less noisy than the relationship between the individual moments and the predictors. Therefore, we avoid the potential misspecification and additional noise due to the estimation of the return distribution as an intermediary step.

1.4 Firm Evaluation

An initial public offering (IPO) is the process of introducing a private firm to a market for the first time. Evaluation of these firms is a challenging task due to the lack of historical information on trading and subsequently insufficient knowledge of their past performance. Regarding IPOs' price performance, an average amount of underpricing (defined as the difference between the close price on the first trading day and the offer price, or equivalently as the initial return), has been documented in some research (Aggarwal et al., 2009; Carter and Manaster, 1990; Miller and Reilly, 1987; Ibbotson et al., 1988; Ritter, 1987; Smith Jr, 1986). For instance, the statistics give positively distributed initial returns overall IPOs or abnormal returns, asserting the existence of an underpricing. Two hypotheses exist that explain the underpricing. The first

one attributes the initial day return to deliberate premarket underpricing of IPOs, which is related to information asymmetry. The other one relates IPO underpricing to mispricing on the first day when trading commences in the stock exchange.

The empirical studies that explore these two hypotheses face an empirical challenge to distinguish deliberate underpricing and aftermarket mispricing from the initial return. This is challenging because the intrinsic value or fair price is unknown, and it is impossible to estimate deliberate underpricing or aftermarket mispricing without knowing the fair offer price. In this thesis, we focus on this issue and propose an approach based on machine learning (ML) to estimate IPO fair price. More specifically, we generalize the frontier production (linear) function in Stochastic Frontier Analysis (SFA) to a nonlinear function to capture nonlinear relations between fair price and pricing and underpricing variables. We then use Artificial Neural Networks (ANN), as a powerful machine learning tool, to estimate the proposed nonlinear model. We further derive and use the relative estimator suggested by Schmidt and Sickles (1984) in our nonlinear setting to estimate the IPO fair price. By combining the neural network modeling tools with stochastic frontier methodology, and integrating them into the relative inefficiency estimator, we propose a solution method for the IPO pricing efficiency problem that is distribution-free, able to model complex dependencies, and can provide new insight into the behavior of the participants in the IPO procedure with a focus on a few numbers of the main drivers of efficiency.

1.5 Model Analysis in Financial Problems

The use of neural networks as a complex statistical tool improves the estimation of optimal decisions, but NN can be considered as a black box that provides little insight into the influence of input variables on final decisions. However, interpretation of statistical models is desired for both academics and practitioners since it helps them understand the models in a more intuitive manner. We adopted and applied three methods from computer science to better understand the impact of the input variables on outputs in the last two studies in Finance. The first method referred to as

'Partial Derivative (PaD)', helps us to determine the relative impact of each variable using the final trained networks. The second one, which is a method of sensitivity analysis referred to as 'Perturb', is an ideal approach to examine the direction and significance of the impact of an independent variable on the output in a nonlinear model out of the bag. The last method is 'Connection Weights' which is similar to the first method and provides relative importance for the input using the final parameters of the network. These methods guide us to determine the effects and associations of explanatory variables and target variables such as optimal portfolio weights in asset allocation and IPO pricing in firm evaluation, as dependent variables in the presence of numerous covariations with complex cross-sectional dependencies among the independent instances or correlated time series.

The remainder of this thesis is organized as follows. The neural networks are briefly introduced in Chapter 2. Chapter 3 presents the model and results for data-driven newsvendor problem with unobservable feature data and evolving states of the system. Chapter 4 provides state-dependent asset allocation based on different performance ratio maximization and then analyses the final wealth for a state-dependent investor who may change his risk profile during investment horizons. Chapter 5 investigates the pricing efficiency of new firms go public and evaluates their characteristics and relative importance of them in a non-linear framework. The new interpretations based on neural networks lead to useful insights into the complex financial environment. The literature review related to each subject is given in its corresponding chapter. Finally Chapter 6 concludes the thesis. Table 1.1 gives an overview of this thesis.

Table 1.1: A summary of the structure of the thesis.

	Model	Main problems	Main methods
Chapter 3	Integrated estimation and optimization for a single-item newsvendor with unobservable feature	Finding optimal order quantity for a single-period inventory system	Hidden Markov modeling by recurrent neural networks
Chapter 4	State-dependent asset allocation with financial time series	Selecting the best performance ratio and finding the optimal portfolio choice	Empirical risk minimization and supervised learning in neural networks
Chapter 5	IPO pricing efficiency using machine learning	Estimating IPO pricing efficiency, deliberate underpricing, and interpreting the effect of firms' characteristics	Non-parametric stochastic frontier analysis using machine learning

Chapter 2

NEURAL NETWORKS WITH APPLICATION TO DATA-DRIVEN METHODS

The Deep Neural Network (DNN) is a machine learning method for a data set that has some patterns. These patterns include all possible relationships between data. Data are usually connected through linear or nonlinear models. These models exhibit dynamic behavior over time. In many cases, some information is accessible and easy to collect. On the other hand, from the available information, there are some dependent and predictable data in which we can observe and use in prediction. In general, DNN maps n-dimensional independent input data to m-dimensional dependent output variables. A DNN thus behaves as a "mapping machine", modeling a function $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, (Rojas, 2013).

A neural network consists of several nodes that are connected, forming a directed graph. Each node/neuron function receives signals from its upstream neurons and passes the aggregate signal to an activation function. The output of the activation function then in turn is passed to the downstream neurons. Activation functions are typically monotone increasing functions that map the set of real numbers to a finite interval e.g. $[0 \quad 1]$. Some commonly used activation functions include the sigmoid and tanh functions. A deep neural network is a neural network with various hidden layers between the inputs and the outputs. Figure 2.1 depicts a symbolic neural net.

Several types of networks exist and contribute to solving various problems. Feed-forward, recurrent, probabilistic, and fuzzy are some such networks. The data that we are interested in modeling in this thesis has a time dimension that is important in understanding the correlated data. The time dimension can be incorporated into

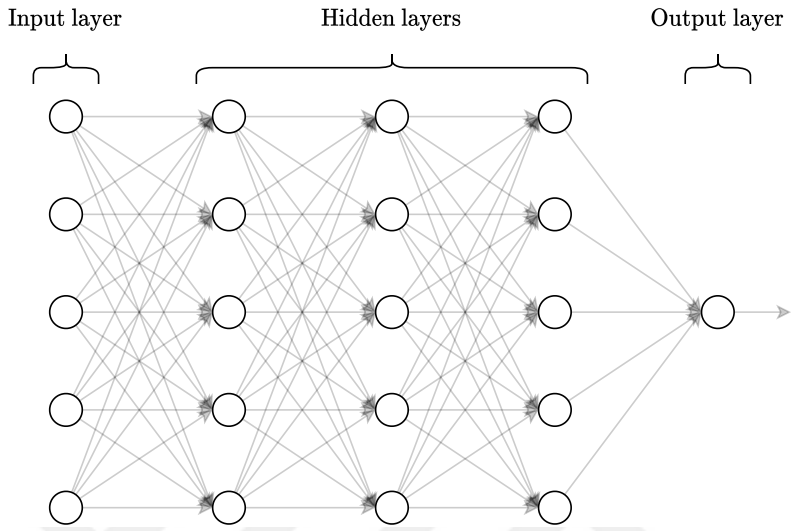


Figure 2.1: A deep neural network.

a neural network by folding a neural network that takes the inputs from different times via different neurons into a network that takes in similar features from different periods, from the same neuron. The folded network is referred to as a recurrent network. In particular, Backpropagation (BP) neural networks are a class of feedforward networks with supervised learning rules. BP networks are the most common multilayer networks estimated to be used in 80% of all applications (Kaastra and Boyd, 1996). In general, designing a network and its implementation involves many steps and parameters. The main steps of *Data preprocessing*, *Topology*, and *Training* are shown in Table 2.1. A BP network contains a set of units of some functions which are connected by directed edges. In addition to these units, there are input units, by which data enters the network. Also, the output units gather all information from the network as final value(s). Units of a kind are located in a layer. In all, there are input layer, an output layer, and hidden layer(s) which include input units, output units, and hidden computing units, respectively.

Table 2.1: Common parameters in designing a backpropagation neural network, (Kaastra and Boyd, 1996)

Data preprocessing
frequency of data - daily, weekly, monthly, quarterly
type of data - technical, fundamental
method of data sampling
method of data scaling - maximum/minimum, mean/standard deviation
Topology
number of input neurons
number of hidden layers
number of hidden neurons in each layer
number of output neurons
transfer function for each neuron
error function
Training
learning rate per layer
momentum term
training tolerance
epoch size
learning rate limit
maximum number of runs
number of times to randomize weights
size of training, testing, and validation sets

The number of input and output units depends on the available data and desired outputs, but the number of hidden layers, computing units, and also the connections between all these types of units, rely on the logic of the problem structure and some criteria which determine the complexity of the network model that prevents the model from overfitting (modeling the error instead of patterns) and underfitting (ignoring the main relationships in modeling).

Computing units are nodes that can receive an unlimited number of inputs while giving out only one value as an output. The information which transits to a node reaches it by a predetermined directed edge(s). According to Figure 2.2, in a node, first, an integration function g which is usually an addition function maps the coming

information of degree, for say n, to a single value, and then another function s named activation function produces the output of the node. The activation functions could be different in each node. Step function, sigmoid, tanh, and many others are some examples of activation functions.

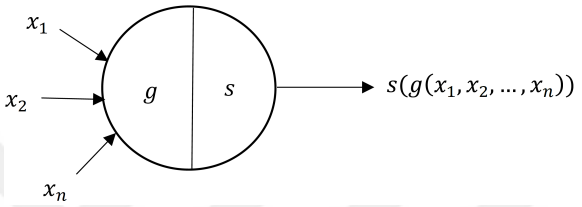


Figure 2.2: A computing unit

Recalling the purpose of the DNN, we are looking for a model on some data to gain desired outputs. The performance of a network with its current parameters is measured by a loss function. So, if assume a prediction problem, a loss function would be needed to compute the closeness of network outputs to the desired ones as targets. To this end, for example, a squared error function is usually used.

In each DNN, the aim is to train the weights of the edges so that the network fits to data as well as possible, Therefore, first, the number of layers, the number of nodes in each layer, the activation function inside each node, and the loss function have to be determined. After selecting those characteristics and building the network, DNN starts with a random initial solution. We provide common and most important considerations in using neural networks for all projects in the following. The specific details are explained in each chapter in separate sections.

2.1 Data Preprocessing

Data scaling is an important thing that has to be implemented before feeding the inputs into the network. It contains methods of data scaling including linear and nonlinear normalization. The former is chosen so that if X is a vector of N features

for all samples, the new normalized data is obtained by the following linear formula:

$$X(i)_{Normalized} = \frac{X(i) - \min(X)}{\max(X) - \min(X)} \ ; \ i = 1, \ldots, N \tag{2.1}$$

The other popular normalization method is to find the z-score of each data point by subtracting each data by mean of the sample μ_i and dividing by the standard deviation of the data σ_i for feature i.

$$X(i)_{Normalized} = \frac{X(i) - \mu_i}{\sigma_i} \ ; \ i = 1, \ldots, N \tag{2.2}$$

2.2 Topology

Designing a DNN is as important as input selection and improving the objective function by the backpropagation algorithm. That is why the *Topology* is listed as one of three main steps in designing a BP neural network in Table 2.1. The number of hidden layers and hidden units is investigated. Several works have developed some Rule of Thumbs (RoT) to choose these parameters optimally. We used two such RoTs which are determined in a multi-layer DNN perceptron. Ke and Liu (2008) approximated the optimal hidden units, M_i, for i^{th} hidden layer with the following formula:

$$M_i = \frac{N + \sqrt{T}}{L} \ ; \ \forall \ i = 1, \ldots, L \tag{2.3}$$

Where L is the number of hidden layers, N is the number of inputs, and T is the number of samples. Another rule of thumb is suggested in Huang (2003) for two-hidden-layer feedforward networks (TLFNs). It is followed by some proofs showing that a TLFN with $2\sqrt{m + 2T}$ ($\ll T$) hidden units can learn any T distinct samples with any arbitrarily small error. Where m is the number of output units. In more

details, the proposed number of hidden units in each hidden layer are as follows:

$$M_1 = \sqrt{(m+2)T} + 2\sqrt{T/(m+2)}$$
$$M_2 = m\sqrt{T/(m+2)}$$

Where M_1 and M_2 denote the number of units in the first and second hidden layer, respectively.

2.3 Training

The training of the ANN is accomplished through a learning process. A neural network is fitted to a given set of data points by changing the weights of the arcs that connect the neurons. The type of learning by which, the model is trained to predict a specified target is referred to as *Supervised learning*. The goal in supervised training a neural network is decreasing the total error of the model in predicting the output variable by changing the network weights. This problem, in general, can be very complicated, however, it has become computationally much less burdensome thanks to the backpropagation (BP) algorithm. The backpropagation algorithm is a gradient-based method that iterates between forward and backward passes through the network modifying the weights of the arcs and calculating the gradients in the network. Before explaining the BP algorithm, we describe our approach to solve problems rather that a simple prediction as the typical task of neural networks. We utilize the principle of empirical risk minimization in the learning theory to support the general approach in different problems.

2.4 Learning Theory and the Application

In this thesis, the main goal in using ANN is to find the optimal outcome as the optimal decision in the intended problems. In the first two projects which are related to the inventory and asset allocation, we propose a decision framework using neural networks to directly prescribe the optimal decision variables. Subsequently, in these

problems, there is no optimal decision as targets to learn form. Instead, we optimize an empirical risk function that ensures that the learning proceeds in the direction of minimizing the distance between the final outputs of the networks as the decision variables and the unobservable optimal ones. However, in the third project, we use the typical mean square error function (MSE) to predict the desired targets. In what follows, we explain the principle of empirical risk minimization that facilitate considering any type of risk function in the learning process.

2.4.1 Empirical Risk Minimization

Optimizing a function using neural networks is a problem of learning that is choosing from a given set of functions, $\Lambda(F, W)$, $W \in \omega$, that one which approximates best the response. In this thesis, this function is a neural network with the parameter W that denotes the network weights and F is a vector of input observations. According to the learning theory proposed by Vapnik (1992), learning corresponds to the problem of function approximation. In order to choose the best approximation function, we need to measure the loss $G = L(y, \Lambda(F, W)$ between the responses provided by the learning machine, $\Lambda(F, W)$, and the actual response, y, associated with a given input F. We utilize the principle of empirical risk minimization (ERM) that states that the risk function G can take any form of loss, risk, or cost function whose simple forms are mean square error or likelihood.

In the first project in chapter 3, function G is a newsvendor cost function in which the order quantity $Q = \lambda(F, W)$ is the output of the machine, while demand D is the response of the supervisor. We obtain the optimal order quantity for each period based on cost minimization and approximating a neural network that provides the best order quantity close to the unobservable optimal one. Similarly, in the second project in Chapter 4, function G is a performance ratio that takes the response of the learning machine as the asset weights $x = \Lambda(F, W)$, while the supervisor's response (actual response) is not the optimal weights x^* but the asset returns R realized after observing the corresponding input variables F. Hence, the loss function is formulated

using a performance ratio as $G(R, \Lambda(F, W))$. ERM assumes that optimal function $\Lambda(F, W^*)$ results in a function $G(R, x^*)$ which has the maximum value. Therefore, in this study, the learning is performed based on the average of ratio values as the empirical risk function using a training set of T observations. Unlike previous two problems, the last problem relates to the simple prediction of offer prices of initial public offerings. The function used in this model is MSE that measures the error of prediction provided by the network and the actual response. In this problem, we develop sub networks to model the stochastic frontier analysis (SFA) method. This need different groups of input data processed into the sub networks by which various elements of SFA model will be attained. Table 2.2 represents the general characteristics of the networks used in different problems of this thesis.

Table 2.2: Networks' characteristics.

	Problems	Data type	Inputs	Output	Network type	Algorithm
Chapter 3	Single-period and single-item newsvendor	Correlated time series	Feature data, such as whether, season, temperature, and so on	Order quantity	Recurrent	Gradient descent
Chapter 4	State-dependent asset allocation	Correlated financial time series	State variables, such as Default, Dividend, Term, Trend	Portfolio weights	Feed-forward	Gradient ascent
Chapter 5	IPO pricing	Independent observations	Firm and deal characteristics, such as the age of a firm and underwriter fee	Offer price and pricing effiecincy	Feed-forward	Gradient descent

In all networks, the BP algorithm is implemented using gradient descent method. This method follows the appropriate direction of the gradients to optimize Function

G. Using this algorithm, we optimize the objective function iteratively by generating a sequence of network weights based on random gradients until there is no improvement in the function. In the next subsection, we briefly describe the fundamental rule of BP algorithm, however it is explained for each problem in more details in each corresponding chapter.

2.4.2 Backpropagation Algorithm

In order to see that how a network works there are two main steps of learning which are appropriate in networks used in this thesis. The Feed-Forward step contains the supply of information into the network, the transition of numerical information from node to node through the edges, and finally gaining outputs from the final layer so as to produce outputs which minimize a loss function. The purpose is to find a method to calculate efficiently the gradient of a multidimensional network function according to the weights of the network. Because the network is equivalent to a complex chain of function compositions, we know the chain rule of differential calculus plays a major role in finding the gradients of the function. As in Figure 2.3, the network is evaluated in two stages: in the first, feed-forward, information comes from the left and each unit evaluates its primitive function s, in its right side as well, as the derivative s' in its left side. Both values are stored in the unit, but only the value from the right side is transmitted to the units connected to the right. The second step, backpropagation, consists of running the whole network backward, whereby the stored values are now used.

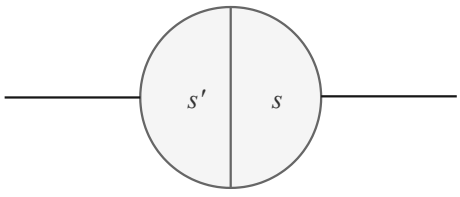


Figure 2.3: The two sides of a computing unit

The first step is straightforward and can be followed from the Equation (2.4). The information between nodes is transferred by the network weights connecting them. Each node receives input values g from previous layer's outputs, and an activation function s provides an output $a = s(g)$ for the node. Consider the weight w_{jk} between the two arbitrary nodes j and k in layers l and $l+1$, respectively. We can write the following to calculate the input for each node k in the layer $l+1$:

$$g_k^{l+1} = \sum_j w_{jk} a_j^l = \sum_j w_{jk} s_j^l(g_j^l) \tag{2.4}$$

Here, the value of the function $s_j^l(g_j^l)$ is the output of the node j and denoted by a_j^l. Generally, all the hidden nodes in a network have a similar activation function s_j^l. Some of the most common functions are sigmoid, tanh, softmax, and linear rectifier. The output of each node is the input of the next layer. The activation function value in the last layer known as the output layer determines the output values of the network.

For the backward step, we should consider the fact that the network is a consecutive substitution of values in activation functions of nodes, thus the partial derivatives could be obtained by using the chain rule. The goal is to compute $\frac{\partial G}{\partial w_{jk}}$. It is obvious from the Equation (2.4) that:

$$a_j^l = s_j^l(g_j^l) = \frac{\partial\, g_k^{l+1}}{\partial\, w_{jk}} \tag{2.5}$$

and we define a backpropagation error for each node computed as follows:

$$\begin{aligned} \delta_j^l &= \frac{\partial\, G}{\partial\, g_j^l} \\ &= \frac{\partial\, G}{\partial\, a_j^l} \frac{\partial\, a_j^l}{\partial\, g_j^l} \\ &= \frac{\partial\, G}{\partial\, a_j^l} (s_j^l)'(g_j^l) \end{aligned} \tag{2.6}$$

To relate δ_j^l and δ_k^{l+1} we can write:

$$\begin{aligned}\delta_j^l &= \frac{\partial G}{\partial g_j^l} \\ &= \sum_k \frac{\partial G}{\partial g_k^{l+1}} \frac{\partial g_k^{l+1}}{\partial g_j^l} \\ &= \sum_k \delta_k^{l+1} \frac{\partial g_k^{l+1}}{\partial g_j^l}\end{aligned} \tag{2.7}$$

From (2.4) we have:

$$\frac{\partial g_k^{l+1}}{\partial g_j^l} = w_{jk}(s_j^l)'(g_j^l) \tag{2.8}$$

Substituting (2.8) to (2.7), we get:

$$\delta_j^l = \sum_k \delta_k^{l+1} w_{jk}(s_j^l)'(g_j^l) \tag{2.9}$$

Finally, we compute gradients of the network with respect to the weights for the main function G by the following equation:

$$\frac{\partial G}{\partial w_{jk}} = \frac{\partial G}{\partial g_j^l} \frac{\partial g_j^l}{\partial w_{jk}} = a_j^l \delta_j^l \tag{2.10}$$

In the next chapters, we apply this general approach to the problems of inventory, asset allocation, and firm evaluation where the structure of each problem is taken into account.

Chapter 3

DATA-DRIVEN NEWSVENDOR PROBLEM WITH UNOBSERVABLE FEATURE

Here, we introduce the single-period newsvendor problem that is considered in this chapter.

3.1 Problem Description

The Newsvendor problem is one of the most well-known inventory models, (Arrow et al., 1951; Silver et al., 1998). This problem is investigated from different points of view. In this research, we examine the aspect of this problem that relates to statistical issues. Similar to operations research and management science (OR/MS) problems, in the Newsvendor problem, a decision maker has to decide under uncertainty. We are interested in understanding the uncertainties, hence, the future by informative data. We consider the simplest form of the Newsvendor where a buyer orders a product for a single period. This is an unconstrained problem where the buyer orders optimally so that his cost will be minimized. The cost function of this problem includes the demand as a random variable. Therefore it should be minimized under uncertainty. Accordingly, in classic Newsvendor, the solution methods focus on the expected value of the cost function. Let $F_D(x)$ denote the cumulative distribution function (CDF) of demand and τ be a ratio of the problem parameters. The inverse of $F_D(x)$ at τ is the minimum point of the objective function. This solution is the order quantity that satisfies the demand optimally. The optimality is based on all periods on average. The cost function and the optimal order quantity for this problem are derived below:

$$C\left(Q,D\right)=h\cdot\left(Q-D\right)^{+}+b\cdot\left(D-Q\right)^{+} \tag{3.1}$$

The optimal order quantity that minimizes $E[C(Q,D)]$ is obtained by some computations as in Equation (3.2),

$$Q^{*}=F_{D}^{-1}\left(\tau\right) \tag{3.2}$$

where $F_D(D) = CDF(D)$ and $\tau = b/(b+h)$.

This solution is applied when the distribution of the demand is assumed to be known. However, this is a strong assumption and the parameters should be estimated by historical data. Recently, efforts have been carried out to solve this problem by data-driven methods. This line of research includes several works that gradually tend to relax the distributional assumptions and draw more information from data. We assume that past data is available on the observable factors and the corresponding demand that was observed. On the other hand, information about the unobservable factor has to be inferred from past data. The traditional approach to such a problem would be to first estimate the underlying model parameters for the effect of the observable factors on demand and infer the hidden state information and its effects. Once estimation and inference take place, the optimization problem would be solved in the second step. However, some of the recent research on this problem (see Ban and Rudin (2019) for example) has demonstrated the advantages of integrating the estimation step of the parameters for the observable factors with the optimization step using tools from machine learning. We pursue this integrated estimation and optimization approach in a model which has the additional complexity of an unobservable factor. We therefore combine estimation, inference and optimization using a multi-layered neural network.

3.2 Literature Review

In the following, we provide a review of the pertinent literature. This is presented in two parts. First, we review the literature on inventory problems with an evolving demand environment. Then, we review the related work on the data-driven newsvendor problem.

3.2.1 Inventory Problems with an Evolving Demand Environment

A long line of research investigates the impact of several factors such as macroeconomic shocks and cycles on inventory systems (Blinder and Maccini, 1991; Shang, 2012; Kesavan and Kushwaha, 2014), especially through the effects of such factors on random demand. Several papers consider dynamic environmental factors that cause the demand distribution to be non-stationary which creates additional challenges. A common assumption in these papers is that the random environment evolves according to a Markov chain.

Earlier studies assume that the state of the Markov chain at each point in time is fully observed and the true demand distribution associated with each state is known. For instance, Feldman (1978) proposes and analyzes a model where demand depends on the state of the environment modeled by a continuous-time Markov process. Lovejoy (1992) investigates the optimality of a myopic policy with non-stationary demand which is dependent on a Markovian process over time. Song and Zipkin (1993) present an inventory model where the demand depends on the state of the world modeled by a Markov chain and derive the optimal ordering policy. Sethi and Cheng (1997) extend the analysis of inventory models with state-dependent demand by taking into account shortage and service level constraints. Beyer and Sethi (1997) derive the average cost optimal ordering policy over an infinite horizon with Markovian demand. Huh et al. (2011) identify the sufficient conditions for the optimality of inventory policies under Markov decision processes. Gallego and Hu (2004) derive optimal policies for production and inventory systems with Markov modulated demand and supply processes.

In many practical situations, the environmental states are not perfectly observable. Instead, one can observe information about the environment and can only infer the states in a probabilistic manner. Treharne and Sox (2002) categorize the literature in terms of stationarity of the demand and observability of the information into four classes. The class of decision systems with Markov modulated demand and partially observed information is known as partially observed Markov decision process (POMDP) (Monahan, 1982). Treharne and Sox (2002) study several inventory policies where only the historical demand is observable and the probability distribution of the demand is determined by the non-observable state of the Markov chain. Bensoussan et al. (2005, 2007) consider the newsvendor problem with censored demand and inventory which depend on the Markov chain states. Arifoğlu and Özekici (2010) analyze a single-item periodic-review inventory system in a random environment. They extend the model of Gallego and Hu (2004) to the more general setting where the environment is only partially observable. In particular, they show that a state-dependent base-stock policy is optimal using sufficient statistics on the environment process. In a later work, Arifoğlu and Özekici (2011) investigate the optimality of a state-dependent inventory policy in a random environment where the capacity of production is random. Avci et al. (2020) model and analyze the inventory problem where the demand belongs to a probability distribution conditional on the Markovian states of the world.

In the above papers with imperfectly observed environment processes, the estimation of the state of the environment is an important subproblem. This subproblem is solved using Bayesian updating and a general solution is given by the Baum-Welch algorithm. The outcome of this algorithm is the estimation of demand distribution for each state of the observed sequence. This estimation is based on maximizing the likelihood of the observed sequence. This maximization at the subproblem level does not take into account the objective function of the inventory problem which leads to a separation of estimation and optimization. However, it is seen in recent examples in the literature that integrating the estimation and optimization problems may lead

to better solutions. We refer to the separated estimation and optimization procedure used in the above papers as the Objective-blind Baum-Welch (ObBW) method. Table 3.1 displays the characteristics of this method along with other approaches that are explained in the next subsection.

3.2.2 Data-driven Approaches to Inventory Problems

Many papers address the concern that the demand distribution in an inventory problem may not be completely known. Particularly, in the earliest studies, the cost function of the newsvendor is optimized with a set of possible distributions for demand. Scarf (1958) proposes a robust optimization approach where the optimal policy is reached by maximizing the minimum expected profit over the set of possible distributions in an uncertainty set. Gallego and Moon (1993) present a new proof of optimality for Scarf's ordering rule and extend the analysis to various other settings. On the other hand, Scarf (1959) considers a Bayesian framework where a parametric approach is taken. The updates, based on observations, then correspond to updating the parameters of the prior distribution. Perakis and Roels (2008) investigate the optimal order quantity to minimize the maximum regret criterion. This is done when partial information (e.g. mean, variance, and symmetry) of the demand distribution is available.

Some studies contribute to relaxations of the assumption that demand distribution is completely known by developing data-based methods. In this framework, the decision maker uses the empirical distribution obtained from past observations. For instance, Levi et al. (2007) investigate an inventory problem using a Sample Average Approximation (SAA) approach, prove its convergence properties, and extend it to a multi-period setting. Liyanage and Shanthikumar (2005) propose a new approach named operational statistics that integrates estimation and optimization and show that this integration improves the solutions when a limited sample of data is available. Other researchers have considered the problem with censored demand data where only sales information is recorded rather than the demand. In this case,

when demand is more than the inventory level, the exact value of the demand is not observable (Levi et al., 2007). Huh et al. (2011) focus on a non-parametric and distribution-free model. They provide theoretical and experimental performances of the obtained policies. Besbes and Muharremoglu (2013) examine the effects of censored demand using a regret-based performance criterion and present results both in the presence and absence of demand censoring.

We refer to the approach in the above papers which is mostly based on the sample observations of demand as the Empirical Demand Distribution (EDD) method. For instance, Bertsimas and Thiele (2005) solve the problem without estimating the distribution but assuming that all of the demand observations in the sample are assigned an equal probability $1/T$, where T denotes the number of demand observations. The optimal stock level or the order quantity is then approximated by the estimated empirical distribution. The advantage of this method is that, unlike the ObBW method, it does not assume any particular shape for the demand distribution. This is useful with real data where demand may not follow a common distribution. On the other hand, one drawback of this method is assuming that all the future observations will also belong to the same empirical distribution which may be questionable.

In recent decades, data-driven optimization under uncertainty has gained increasing attention. For instance, He et al. (2012) model the problem of setting nurse staffing levels in hospital operating rooms with the uncertainty of daily workload as a newsvendor problem. They present various models including a linear decision model that uses two features. Sachs (2015) considers ordering with different types of exogenous data such as price and temperature that might explain demand. He formulates the optimal inventory level as a linear function of those variables. In a case study with real data, he shows that the non-parametric approaches outperform the parametric ones. This is advantageous when the true demand distribution is not completely known and several exogenous variables are available. We refer to this approach as Parameter Fitting Linear Regression (PfLR). Here, the exogenous features explain part of demand variability through a (typically) linear relationship. The

approach, therefore, estimates the mean of the demand by linear regression on the features. This results in a time-varying mean that depends on the features and a fixed standard deviation. In addition, the usual Gaussian assumptions are usually taken. Similar to the empirical method, this method is not able to capture the dependency between demand observations.

The integrated estimation and optimization approach is referred to as a prescriptor method (Van Parys et al., 2020). In a recent paper, van der Laan et al. (2019) propose a new data-driven approach based on distributionally robust optimization to achieve on-target service levels. They show that the suggested approach, which bases the inventory decision directly on feature data, is more reliable than several classical approaches even with a limited number of historical observations. Several studies combine the estimation and optimization steps using tools from machine learning. Ban and Rudin (2019) consider the newsvendor problem with n observations and p features in two cases of a low and a high number of features. They propose two Machine Learning-based approaches: regularization and Kernel Optimization (KO), and demonstrate some theoretical properties. They also show, in a numerical study, that using such features may lower the expected cost significantly. A recent paper by Khayyati and Tan (2020) shows that integrating the two steps of parameter estimation and optimization can improve the performance of a system in make-to-stock queues.

The general use of ML-based methods in joint estimation and optimization, however, goes back to several earlier studies such as (Efendigil et al., 2009; Goel et al., 2010; Gruhl et al., 2004, 2005). Bertsimas and Kallus (2020) combine the methods of ML with the conditional stochastic optimization problem. They include direct-effect data as well as other auxiliary information and assume that the joint probability distributions are unknown and the observations are imperfect. They develop the framework with several ML methods and show that these techniques are computationally tractable and asymptotically optimal under some conditions. This tractability is shown in the presence of dependencies in the data and censored observations.

The main contributions of these important recent papers are using informative

data, benefiting from non-parametric models, decreasing the estimation errors, and making the decisions more dynamic. We categorize the approach proposed by Ban and Rudin (2019) of relating the optimal order quantity directly to the features using a functional form for the relationship as Linear Machine Learning (LML). This method combines the estimation and optimization steps by solving a nonlinear optimization problem, where the objective is minimizing directly the cost function of the newsvendor problem instead of minimizing the regression error. In this method, similar to the parameter fitting approach, there is an assumption of the linear relations between features and the optimal order quantity.

Finally, some recent studies in data-based optimization contribute to the literature by taking nonlinear relationships between feature data and the order quantity into consideration. Oroojlooyjadid et al. (2020) apply a deep learning approach to the newsvendor problem when such non-linearities exist. They also consider multi-feature and multi-products extensions of the problem. It is shown that deep learning outperforms other benchmarks such as local regression, classification and regression trees, random forests, and kernel optimization, especially when demand is highly volatile. Seubert et al. (2020) develop a data-driven system of ordering for a bakery chain based on artificial neural networks. They use two different methods of sequential and joint estimation and optimization and show that both methods considerably save costs compared to human planners. Qi et al. (2020) extend the approach of Oroojlooyjadid et al. (2020) to a multi-period inventory system with uncertain demand and vendor lead time. Zhang and Gao (2017) examine a supervised deep learning algorithm with two objectives. They demonstrate that the original newsvendor loss function as the training function outperforms the quadratic loss function. The algorithm has been evaluated on synthetic and real data. In this class of methods, non-linear relations between features and demand have been considered. Oroojlooyjadid et al. (2020), as the first study of this class, suggest complex functions that relate the features and the optimal order quantity using deep learning in the classical newsvendor problem. This method can use the features and optimize the cost function while considering a wide

range of relationships in addition to linear relations. However, their method does not identify the dependencies between consecutive states of the world in an evolving environment.

3.2.3 Benchmarks

In the following, we list the benchmark methods and explain more specifically how they solve the intended newsvendor problem.

EDD: The empirical demand distribution approach uses only the demand observation and finds the optimal solution based on the empirical distribution which assumes equal weights for each observation. The optimal quantity is obtained by the known formula.

ObBW: The Objective-blind Baum-Welch algorithm considers the Markov chain which subordinates the demand (Avci et al., 2020; Treharne and Sox, 2002). Consequently, it finds the most probable sequence of states by the predetermined distribution for demand. Therefore, this is a parametric approach that fits a distribution on demand and estimates the appropriate parameters for each state of the Markov model. Baum-Welch algorithm is a well-known forward-backward method in the estimation of hidden Markov states using observable data (i.e. demand). Details of this algorithm are described by Rabiner (1989).

PfLR (OLS): The parameter fitting linear regression is the first and the simplest method which takes the feature data into account. It consists of two steps: estimating the parameters of the demand distribution and then use the estimations in optimization problem (Ban and Rudin, 2019). However, this approach ignores the dependency between demand observations through the hidden Markov states. That said, it is a dynamic approach that makes a linear relationship between features and demand by

the following linear regression

$$D_t = \beta_0 + \beta_1 f_{1t} + \beta_2 f_{2t} + \ldots + \beta_N f_{Nt} + \epsilon_t, \tag{3.3}$$

where $t = 1, \ldots, T$, and N is the number of features. The time-varying mean for demand and the standard deviation of the estimated normal distribution is calculated by the coefficients and residuals of the regression in (3.3). The standard deviation of the demand distribution is equivalent to the standard deviation of the residuals. The optimal order quantity would be

$$Q_t^* = \beta_0 + \beta_1 f_{1t} + \beta_2 f_{2t} + \ldots + \beta_N f_{Nt} + z^* \sigma_\epsilon, \tag{3.4}$$

where $z^* = \Phi^{-1}(b/(h+b))$.

LML: Ban and Rudin (2019) suggest a linear decision rule for the order quantity as

$$Q = \left\{ q : \mathcal{X} \to \mathbf{R} : q_t(\beta) = \beta_0 + \sum_{n=1}^{N} \beta_n f_{nt} \right\}. \tag{3.5}$$

They substitute this linear formula into the newsvendor problem and solve a nonlinear program for the optimal order quantity.

DNN: Oroojlooyjadid et al. (2020) apply a deep neural network to the newsvendor problem which is a nonlinear extension to the LML model. In this approach, a neural network maps the feature data to the optimal decision. The goal of the network is to provide the minimum average cost value over all periods

$$\min_{\mathbf{W}} \ \frac{1}{T} \sum_{t=1}^{T} NVC(\theta(\mathbf{F}_t, \mathbf{W}), D_t), \tag{3.6}$$

where $\mathbf{W}$ is the matrix of the network weights, $\mathbf{F}_t$ is the vector of input features at time t, and the network is indicated by the mapping function θ. To have a network

Table 3.1: Different approaches to solve Newsvendor problem

	Method	Dependency	Features	NV cost function	Non-linearity
1	Linear HMMNV	✓	✓	✓	
2	Non-linear HMMNV	✓	✓	✓	✓
3	Empirical Demand Distribution			✓	
4	Objective-blind by Baum-Welch Alg.	✓			
5	Parameter fitting (Linear regression)-OLS		✓		
6	Linear Machine Learning (Rudin and Vahn, 2014)		✓	✓	
7	Deep Learning (Oroojlooyjadid et al., 2020)		✓	✓	✓

structure in the DNN method similar to the HMMNV method, we use the same hyperparameters selecting rule (network specification including the number of hidden layers and the number of hidden nodes in each layer) explained for the HMMNV method.

In this study, we propose a new algorithm that utilizes the machine learning method of Deep Neural Networks (DNN) and amends the shortcomings associated with the approaches explained in the literature. The proposed methods referred to as Linear HMMNV and Non-linear HMMNV are explained in two forms of relationships between data. In table 3.1, we show how the new methods amend the shortcomings.

Sections 3.3 and 3.6 represent the suggested methods.

3.3 Linear HMMNV

We consider a single item Newsvendor problem where in every period, before the realization of the demand, a set of features are observed. In addition, the state of a

hidden Markov model influences the distribution of the demand at each time. The data of the problem can be represented by tuples of feature vectors and demand realizations as given in Equation (3.78), where T denotes the number of periods.

$$\mathcal{D} = \{(\boldsymbol{F}_1, D_1), (\boldsymbol{F}_2, D_2), \ldots, (\boldsymbol{F}_T, D_T)\} \tag{3.7}$$

where $\boldsymbol{F}_t$ for $t = 1, 2, \ldots, T$ are identically independent distributed. The vector of $\boldsymbol{F}_i$ for sample i consists of N different feature data $(f_1, f_2, \ldots, f_N)$.

3.3.1 Special Case: a Newsvendor Problem with Markov Modulated Demand and Observable States

Many papers in the literature assume that demand depends on an external state of the world that evolves according to a Markov chain S. It is then natural to assume that demand D depends on the external state and therefore the demand in period t, $D_t = D|S_t$ (or $D|S_{t-1}$ depending on the filtration). We then look for the order quantity that minimizes the expected cost of the system as

$$\min_{\mathbf{Q}} \; NVC(Q) = E_D\left[h(Q-D)^+ + b(D-Q)^+|S\right], \tag{3.8}$$

where D denotes the random demand, Q is the order quantity, and h and b denote the unit holding and lost sale costs, respectively. Given the state information, with full knowledge about the parameters of the demand distribution or having the estimation using historical data, the optimal order quantity can be obtained by solving the above optimization problem for a period t:

$$Q_t^* = F_{D|S_t}^{-1}\left(\frac{b}{h+b}\right), \tag{3.9}$$

where $F()$ is the cumulative distribution function of demand.

Let us now generalize the model further and define a base demand B_i that depends on $S_t = i$ and an additional demand that depends on the values of certain observed

features $f_{1t}, f_{2t}, \ldots, f_{Nt}$ that do not depend on S_t. Further, let us assume that B_i is independent of features. We can then have

$$D_t = B_i + \psi(f_{1t}, f_{2t}, \ldots, f_{Nt}) + \epsilon_t, \tag{3.10}$$

where ϵ_t is a random error term with $E[\epsilon_t] = 0$.

Example

As an example, assume that there are two states of the world: (1) Good and (2) Bad, and B_i is normally distributed with parameters (μ_i, σ_i) where $i = (1), (2)$. Further assume that

$$\psi(f_{1t}, f_{2t}, \ldots, f_{Nt}) = \beta_0 + \beta_1 f_{1t} + \beta_2 f_{2t} + \ldots + \beta_N f_{Nt}. \tag{3.11}$$

We then have that D_t is normally distributed with mean $\mu_i + \psi(f_{1t}, f_{2t}, \ldots, f_{Nt})$ and variance $\sigma_i^2 + \sigma_\epsilon^2$.

From the known results, we then have

$$Q_t^* = \mu_i + \psi(f_{1t}, f_{2t}, \ldots, f_{Nt}) + z^* \sqrt{\sigma_i^2 + \sigma_\epsilon^2}, \tag{3.12}$$

where $z^* = \Phi^{-1}(b/(h+b))$ (and $\Phi()$ is the cumulative distribution function of a standard normal random variable).

We present the main model in this chapter which includes an unobservable environment process states.

In this section, we consider a linear relationship between features and demand. The realizations of the features are independent from each other and the period. Demand is assumed to come from different distributions. The demand distribution depends on some states which have two properties; first, follow a Markov model, second, are hidden. The model which produces the time series of states is a Markov model. The state of the Markov chain is observable only through the demand realization. The

probability densities are defined by some other data as features beside of demand. Therefore, the joint probability distribution of demand and a state varies depending on some relations that explain the demand in each state. An unknown relation is considered between D_t and $\boldsymbol{F}_t$. This relation is shown by the linear function as in Equation (3.13) where $W = (w_0, \ldots, w_N)$ is a set of model parameters corresponding to state s at time t:

$$D_t = \sum_{j=1}^{N} w_j^s f_{tj} + \epsilon_t \quad , \quad for\ t = 1, \ldots, T\ ;\ s \in S \tag{3.13}$$

and ϵ_t is the error term and has a normal distribution; hence $\epsilon_t \sim \mathcal{N}(0, \sigma_{s_t})$ is a positive number. Figure 3.1 shows how demand is produced by features and states.

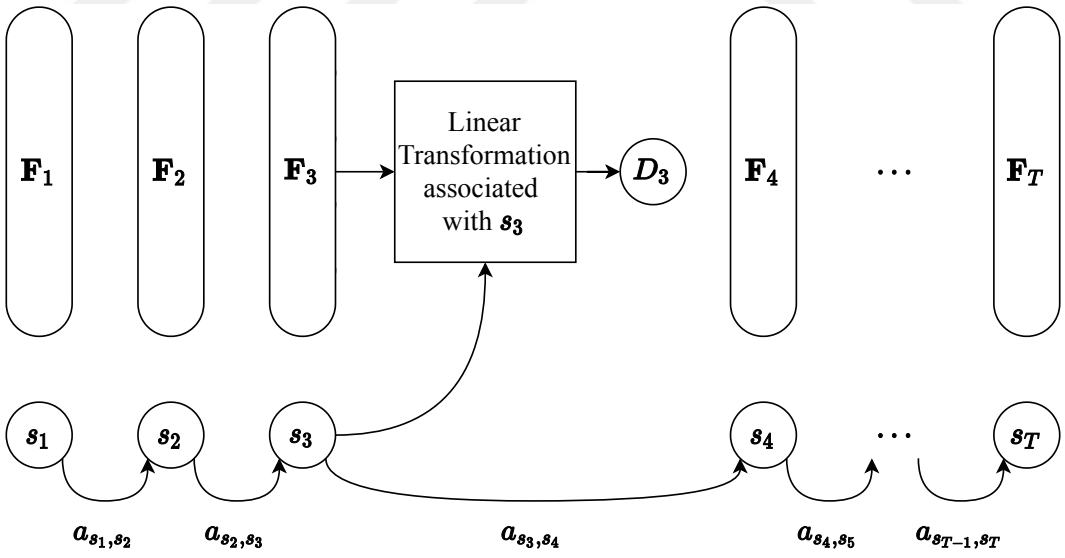


Figure 3.1: The evolution of the state and its effect on the demand

3.3.2 Our Model: a Data-Driven Newsvendor Problem with Unobservable Environment Process

In an inventory system, there may be several sources of uncertainty, generated by observable sources as features and non-observable sources. The features may correspond to observations such as the weather, seasonality, and local market conditions. In this model, the base demand takes place as the non-observable source. The base demand distribution, as a part of whole demand, depends on some evolving states which have two properties; first, they follow a Markov model, second, they are not observable. The state of the Markov chain affects the system partially through the base demand. Therefore, the joint probability distribution of demand and a state varies depending on the base demand distribution in each state, and the feature-dependent part completes the demand distribution independently of the states. Let us assume that S_t is not observable but can only be inferred from past demands $D_1, D_2, \ldots, D_{t-1}$. This is similar to the setup in Treharne and Sox (2002) and Arifoğlu and Özekici (2010). One can then estimate the conditional distribution

$$\hat{S}_t = S_t | D_1, D_2, \ldots, D_{t-1}. \tag{3.14}$$

In our model, we assume that the realizations of the features are independent of each other and the time. Any dependency in the sequence of the observations of a feature and any dependency between the different features do not add more information to the decision at the beginning of a period as the features are observed before setting the order quantity. Hence, the presence of these dependencies does not affect the solution. Moreover, a dependency between the states of the (unobservable) Markov chain and the features may be beneficial to the order quantity decision as the state of the Markov chain can be better inferred through the feature observations

$$\hat{S}_t = S_t | D_1, D_2, \ldots, D_{t-1}, \mathbf{F}_{t-1}. \tag{3.15}$$

3.3.3 Linear Predictive Hidden Markov Model

To investigate the statistical model for the data described above, we modify the Linear Predictive Hidden Markov Model (LPHMM) proposed by Poritz (1982) who considered a time series of length $T \times M$ which is decomposed into a sequence of shorter segments of length M. Each segment is represented by a local model as output of one of S previously selected all pole recursive filters driven by previously selected noise sources. The filters are defined by polynomials of some degree N with $N < M$. What follows are the algorithm proposed by Poritz (1982).

3.3.4 The Local Model: Predictive Densities

Let $Y = (\boldsymbol{F}, D) \in \Re^{N+1}$ be any vector of features and demand at a given time; set a new form of it below:

$$U(Y) = \begin{pmatrix} D \\ f_1 \\ \vdots \\ f_N \end{pmatrix} \tag{3.16}$$

and let

$$R(Y) = U(Y)\, U(Y)^* \tag{3.17}$$

where $*$ denotes transpose. Knowing that the multiplication of a matrix to its transpose is a symmetric positive semi-definite (SPSD) matrix. Therefore, $R(Y)$ is an $N+1$ by $N+1$ SPSD matrix. Let $W = (w_0, \ldots, w_N)$ be the vector of coefficients in real $N+1$ space $\Re^{N+1}$.w_0 corresponds to the demand and is equal to 1. Suppose that the demand D would be predicted by a linear transformation of features so that $\tilde{D} = -\sum_{i=1}^{N} w_i f_i$, then $WR(Y)W^* = \left(D - \tilde{D}\right)^2$. This value is the error of the prediction of demand by features. Let σ be a positive number. A real function is defined

in Poritz (1982) as following:

$$L\left(Y|W,\sigma\right)=\frac{2}{\sqrt{2\pi\sigma^{2}}}\exp\left(-WR\left(Y\right)W^{*}/2\sigma^{2}\right) \tag{3.18}$$

In this function, errors are measured that are obtained by mapping features to demand induced by W and in σ units. This index indicates how well the demand is predicted by features. This function is called a *predictive or recursive density* determined by the model (W,σ). In general, we look for a pair of (W,σ) with $W\in\Re^{N+1}$ and $\sigma>0$ that maximizes this function. In the following, the proof of existence and uniqueness of the maximization point is provided by Poritz (1982):

For $V=(v_0,v_1,\ldots,v_N)\in\Re^{N+1}$ set $V_N=(v_1,\ldots,v_N)$ so that $V=(v_0,V_N)$. In relation to this decomposition $\Re^{N+1}=\Re\oplus\Re^{N}$, any $N+1$ by $N+1$ symmetric matrix can be decomposed to R:

$$R=\begin{pmatrix} B & C \\ C^{*} & D \end{pmatrix} \tag{3.19}$$

where D is an N by N matrix.

Lemma 3.3.1. *(Poritz, 1982) Given a positive definite symmetric $N+1$ by $N+1$ matrix, R, define the smooth real value function F with domain $\Re^{N}\times(0,\infty)$ by the formula: $F\left(W_N,\sigma\right)=\log\left(\sigma\right)+\frac{WRW^{*}}{2\sigma^{2}}$ where $W=\left(1,\hat{W}_N\right)$. Then F has a unique minimum and this occurs at the point $(\hat{W}_N,\hat{\sigma})$ where $\hat{W}_N=-CD^{-1}$, $\hat{W}=\left(1,\hat{W}_N\right)$ and $\hat{\sigma}^{2}=\hat{W}R\hat{W}^{*}$.*

Proposition 3.3.1. *Let $Y\in\Re^{N+1}$ be a vector of features and demand, and suppose that $R\left(Y\right)$ is nonsingular. Then $L\left(Y|W,\sigma\right)$ has a unique maximum and this occurs at $(W,\sigma)=\left(\hat{W},\hat{\sigma}\right)$ where $\hat{W}_N=-C\left(Y\right)D(Y)^{-1}$, $\hat{W}=\left(1,\hat{W}_N\right)$, and $\hat{\sigma}^{2}=\hat{W}R\hat{W}^{*}$.*

3.3.5 The Global Model

Consider a Markov chain model with a discrete time and discrete states. It has S states which is an integer number. The initial probabilities of states are specified by

a vector of $(\pi_1, \pi_{2,}..., \pi_S)$. Each element corresponds to each state that a sequence might start with. The transition probabilities between states are defined by elements of a transition matrix as follow:

$$\begin{pmatrix} a_{11} & a_{12} & a_{1S} \\ a_{21} & a_{22} & a_{2S} \\ \vdots & \vdots & \vdots \\ a_{S1} & a_{S2} & a_{SS} \end{pmatrix}$$

Since the components in this matrix are in probability, they are nonnegetive as well as sum to one in each row. If G is a set of observations in some states of a Markov chain, a transformation of G under a function L along with S forms a set of output densities. These values sum to 1 for each state. Let the function $L_s(Y) = L(Y|W(s), \sigma(s))$ be as defined previously. This implies a linear hidden Markov model that a set of observations map to a value by a linear transformation.

In the setting of a hidden Markov model, let Λ be a set of possible parameters and $\lambda = \{a_{sr}, W(s), \sigma(s) | s, r \in S\}$ be one of them. A time series is considered with a sequence of observations $\mathcal{Y} = (Y(0), \ldots, Y(T))$ of length T. Let Ω be the set of all paths from the state space $\{s_1, \ldots, s_T\}$. Given a believed set of parameters of λ, the probability of a sequence of states ω is the likelihood of observing such observations. First, the probability of ω given λ by the Markov chain is $P(\omega|\lambda) = \pi_{s_1} \prod_{t=1}^{T-1} a_{s_t s_{t+1}}$. Then the likelihood of $\mathcal{Y}$ given the ω and λ is:

$$P(\mathcal{Y}|\, \omega, \lambda) = \prod_{t=1}^{T} L(Y(t) | W(s_t), \sigma(s_t)) \tag{3.20}$$

As $L(\mathcal{Y}, \omega|\lambda) = L(\mathcal{Y}|\ \omega, \lambda)\, P(\omega|\lambda)$ the likelihood of observation over all possible state sequences is:

$$\begin{aligned} L(\mathcal{Y}|\lambda) &= \sum_{\omega\in\Omega} L(\mathcal{Y}, \omega|\lambda) \\ &= \sum_{\omega\in\Omega} \prod_{t=1}^{T} a_{s_{t-1}s_t} L(Y(t)\,|W(s_t), \sigma(s_t)) \end{aligned} \tag{3.21}$$

Solving (3.21) is extensive as it is exponential in T. An efficient algorithm to this issue is proposed by Baum et al. (1970) named Forward-Backward algorithm. It diminishes the complexity of the solution and makes it linear in T.
In Forward-Backward algorithm, two terms of α and β are defined for each state at each time. Let the forward term be $\alpha_0(s) = a_s L_s(Y(0))$, $s = 1, \ldots,\ S$ and

$$\alpha_t(s) = \sum_{r=1}^{S} \alpha_{t-1}(r)\, a_{rs} L_s(Y(t)) \tag{3.22}$$

for $s = 1, \ldots,\ S$ and $T = 1, \ldots,\ T-1$. The backward term is $\beta_{T-1}(s) = 1$, $s = 1, \ldots,\ S$ and

$$\beta_t(s) = \sum_{r=1}^{S} \beta_{t+1}(r)\, a_{sr} L_r(Y(t+1)) \tag{3.23}$$

for $s = 1, \ldots,\ S$, and $T = T-2, \ldots,\ 0$. These recursive formulas are derived with this fact that:

$$\alpha_t(s) = P_\lambda(Y(0), \ldots,\ Y(t), S_t = s) \tag{3.24}$$

$$\beta_t(s) = P_\lambda(Y(t+1), \ldots, Y(T) \mid S_t = s) \tag{3.25}$$

so they satisfy $\alpha_t(s)\,\beta_t(s) = L(\mathcal{Y}, s_t = s|\lambda)$. This the likelihood of observations so that the state at time t is s. Simply, the likelihood of all observations given the model λ is $L(\mathcal{Y}|\lambda) = \sum_{s=1}^{S} \alpha_T(s)$.

3.3.6 Global Maximum Likelihood

We would like to find a model λ that renders the maximum likelihood of a given time series of $\mathcal{Y}$ among all models in Λ. Since there is no closed form solution for hidden Markov models, this problem could be solved by an iterative hill climbing technique. By using the Lemma stated previously, an extension of that method is represented to the class of LPHMM. To this end, the posterior probability of state s at time t based on $\mathcal{Y}$ and λ which is evaluated as follows is required:

$$\gamma_t(s) = \alpha_t(s)\ \beta_t(s) / L(\mathcal{Y}|\lambda) \tag{3.26}$$

for $s = 1, \ldots, S$ and $T = 1, \ldots, T-1$. To find the transition probabilities sort of similar concepts are defined as below:

$$\gamma_t(s,r) = \alpha_t(s)\ a_{sr} L_r(Y(t+1))\ \beta_{t+1}(r) / L(\mathcal{Y}|\lambda) \tag{3.27}$$

for $r, s = 1, \ldots, S$ and $T = 1, \ldots, T-2$. Now, as in Poritz (1982) and regarding Equation (3.17), the local optimization is incorporated into the global form of the linear hidden Markov. Thus the new matrices are $R(Y(t)) = U(Y(t))\, U(Y(t))^*$ for $T = 0, \ldots, T-2$ and for each state defining:

$$R(s) = \sum_{t=0}^{T-1} \gamma_t(s)\, R(Y(t)) \tag{3.28}$$

Similar to the decomposition in Equation (3.19), for $R(s)$ this will be:

$$R(s) = \begin{pmatrix} B(s) & C(s) \\ C(s)^* & D(s) \end{pmatrix} \tag{3.29}$$

where each $D(s)$ is an N by N matrix. Finally define:

$$R(\mathcal{Y}) = \sum_{t=0}^{T-1} R(Y(t)) \tag{3.30}$$

Theorem 3.3.1. *Suppose $\mathcal{Y}$ is a time series with T samples and $R(\mathcal{Y})$ is nonsingular. For any interior model $\lambda = \{ a_{sr},\ W(s),\ \sigma(s) | s, r \in S\}$ define a new model $\hat{\lambda} = \{\hat{a}_{sr}, \hat{W}(s), \hat{\sigma}(s) | s, r \in S\}$ by:*

$$\hat{a}_{sr} = \sum_{t=0}^{T-2} \gamma_t(s,r) \Big/ \sum_{t=0}^{T-2} \gamma_t(s) \tag{3.31}$$

$$\hat{W}_N(s) = -C(s) D(s)^{-1}, \quad \hat{W}(s) = \left(1, \hat{W}_N(s)\right) \tag{3.32}$$

$$\hat{\sigma}(s)^2 = \hat{W}(s) R(s) \hat{W}(s)^* \Big/ \sum_{t=0}^{T-1} \gamma_t(s) \tag{3.33}$$

Then $\hat{\lambda}$ is an interior model and if $\hat{\lambda}$ differs from λ then $L\left(\mathcal{Y}|\hat{\lambda}\right) > L(\mathcal{Y}|\lambda)$

The inevitability of matrices $D(s)$ is a consequence of singularity of $R(\mathcal{Y})$ and the fact that λ is an interior model. The theorem gives an iterative algorithm that each output model is the next input model. Iteration continues until a maximum number of iterations or a convergence criterion like a minimum tolerance on a variable is met.

3.3.7 Data-Driven Newsvendor Problem in DNN

Oroojlooyjadid et al. (2020) applied Deep Learning to the Newsvendor problem. They investigated the problem from two aspects. Multi-product and multi-feature are two extensions of this problem in their work. It was shown that deep learning outperforms other approaches such as data-driven and machine learning approaches, especially with high volatility demand. Zhang and Gao (2017) examined a supervised deep learning algorithm with two objectives. They demonstrated that the original Newsvendor loss function as the training function outperforms the quadratic loss function. The algorithm has been evaluated on synthetic and real data.

In this research, the main structure of the algorithm and consequently the neural network is similar to the recent works, Oroojlooyjadid et al. (2020) and Zhang and Gao (2017). The difference comes back to the inputs. Inputs contain the information of states besides other features when introducing time-dependent demands following a hidden Markov model. Now let consider a data-driven single item Newsvendor

problem which has some feature data $(f_1, f_2, \ldots, f_N)$ as inputs of the network, and the order quantity Q as its single output. Then we may have a number of T sample pairs of data like $\{(\boldsymbol{F}_1, D_1), (\boldsymbol{F}_2, D_2), \ldots, (\boldsymbol{F}_T, D_T)\}$. The vector of $\boldsymbol{F}_i$ for sample i consists of N different feature data. For now to construct the network we consider one hidden layer with M computing units. The activation function in each computing unit is a sigmoid function. The sigmoid function is a continuous and differentiable function allowing to compute the derivatives of the network efficiently. To evaluate the performance of the network outputs we consider the original Newsvendor cost function as the loss function for this network. In order to calculate the derivatives for backpropagation algorithm, the parameters and the flow of the information are specified by notations more precisely. In this DNN, there are an input layer with N units, one output unit, and one hidden layer with M units. These layers require two sets of weights that connect units to each other. One between input layer and hidden layer and the other between hidden layer and output unit. $W^{(1)}$ and $W^{(2)}$ indicate these weights respectively. The information which reach unit i in the hidden layer and single output unit is shown with $net_i^{(1)}$ and $net^{(2)}$ respectively. Once hidden unit i computes its output by activation function, the output of that unit exits from it and is shown by $o_i^{(1)}$. It is seen that $o_i^{(1)} = s\left(net_i^{(1)}\right)$, and in output unit the activation function is a simple addition function so simply the output of this node is identical to the transmitted value to it and they also represent the order quantity $Q = o^{(2)} = net^{(2)}$.

If we consider a sample being fed into the network, we could drive the calculations for it, and then extend them for all samples in a matrix form. At the end of the feed-forward step, once we get the output from the network in the output node, the corresponding Newsvendor cost will be computed by the Newsvendor cost function as in the most right node in Figure 3.2, where h and b are holding cost and back order cost respectively.

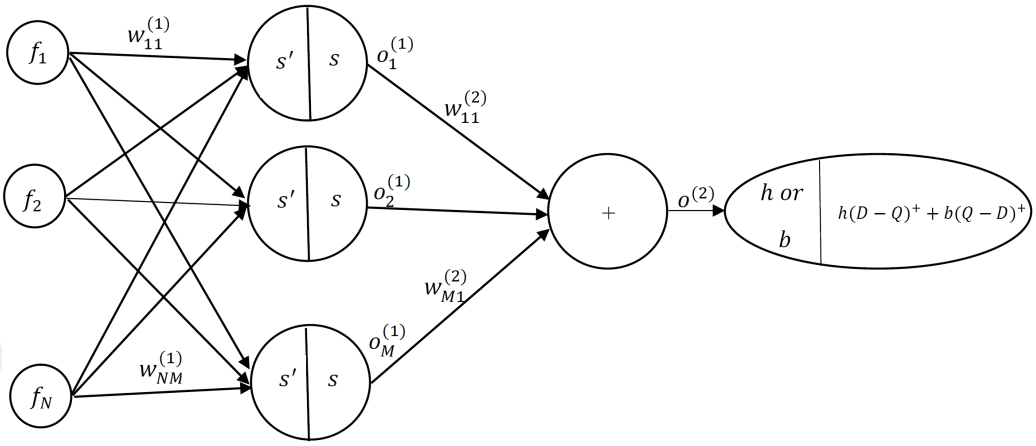


Figure 3.2: A neural network for newsvendor.

Backpropagation step begins right after feed-forward step. In this step we are looking for two sets of partial derivatives $\frac{\partial E}{\partial w_{ij}^{(1)}}$ and $\frac{\partial E}{\partial w_{ij}^{(2)}}$. So, we need to compute the derivatives of activation functions in each node and find the path to a certain edge. This takes us to the intended partial derivative. The derivative of the Newsvendor cost function E w.r.t. the output Q or equivalently $o^{(2)}$ is as follows:

$$\frac{\partial E}{\partial o^{(2)}} = \begin{cases} h & if\ D < Q \\ b & if\ Q < D \end{cases} \tag{3.34}$$

As the activation function in the output unit is a simple additional function, the derivative of it w.r.t. its input value, $\frac{\partial o^{(2)}}{\partial net^{(2)}}$ is just equal to 1. In hidden layer units, sigmoid function has a property that its derivative results in an expression that consists only the function itself. So if the output of hidden unit j is $o_j^{(1)}$, the derivative would be $o_j^{(1)}(1 - o_j^{(1)})$. Now for simplicity of chain rule in taking the derivatives we define a term as backpropagation error for hidden and output layers with $\delta_j^{(1)}$ for $j = 1, 2, \ldots, M$ and $\delta^{(2)}$ respectively. We are moving from the right of the network to the left, so firstly we can compute partial derivatives of $\frac{\partial E}{\partial w_{j1}^{(2)}}$. To this end we define

backpropagation error of output unit:

$$\delta^{(2)} = \frac{\partial E}{\partial o^{(2)}} \frac{\partial o^{(2)}}{\partial net^{(2)}} \tag{3.35}$$

In this problem $\delta^{(2)}$ would be:

$$\delta^{(2)} = \begin{cases} \delta^{(2)}(h) = h.1 & if\ D < Q \\ \delta^{(2)}(b) = b.1 & if\ Q < D \end{cases} \tag{3.36}$$

In Figure 3.3, it can be seen that based on an edge with weight $w_{j1}^{(2)}$ which connects the j^{th} hidden unit to the single output unit, how the chain rule continues to find the partial derivative of $\frac{\partial E}{\partial w_{j1}^{(2)}}$ using this figure.
Now, one more multiplication left to get to the partial derivative $\frac{\partial E}{\partial w_{j1}^{(2)}}$ which is $\frac{\partial net^{(2)}}{\partial w_{j1}^{(2)}} = o_j^{(1)}$ Finally we have:

$$\frac{\partial E}{\partial w_{j1}^{(2)}} = \delta^{(2)} o_j^{(1)} \tag{3.37}$$

which is here:

$$\frac{\partial E}{\partial w_{j1}^{(2)}} = \begin{cases} \delta^{(2)}(h) o_j^{(1)} & if\ D < Q \\ \delta^{(2)}(b) o_j^{(1)} & if\ Q < D \end{cases} \tag{3.38}$$

The next set of partial derivatives $\frac{\partial E}{\partial w_{ij}^{(1)}}$ are the derivatives of cost function w.r.t. the weights between input units and hidden units. In order to compute the back-propagation error $\delta_j^{(1)}$ of unit j in the hidden layer, all possible backward paths which

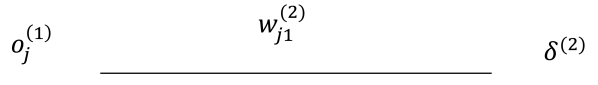


Figure 3.3: Input and backpropagated error at an edge

reach to hidden unit j should be considered so that an integration weighted of backpropagation errors of output layer $\delta^{(2)}$ will transit to hidden unit j. Since here there is only one edge between each hidden unit and the single output unit, transition of backpropagation error to hidden unit j is $w_{j1}^{(2)}\delta^{(2)}$. Similar to the previous set of derivatives, a term of $\frac{\partial o_j^{(1)}}{\partial net_j^{(1)}} = o_j^{(1)}(1-o_j^{(1)})$ should be multiplied by $w_{j1}^{(2)}\delta^{(2)}$. Then backpropagation error for unit j in the hidden layer is calculated as follows:

$$\delta_j^{(1)} = o_j^{(1)}\left(1-o_j^{(1)}\right) w_{j1}^{(2)}\delta^{(2)} \tag{3.39}$$

Again in this problem we have two types of backpropagation error:

$$\delta_j^{(1)} = \begin{cases} \delta_j^{(1)}(h) = o_j^{(1)}\left(1-o_j^{(1)}\right)\ w_{j1}^{(2)}\delta^{(2)}(h) & if\ D<Q \\ \delta_j^{(1)}(b) = o_j^{(1)}\left(1-o_j^{(1)}\right)\ w_{j1}^{(2)}\delta^{(2)}(b) & if\ Q<D \end{cases} \tag{3.40}$$

According to the chain rule, to obtain the partial derivative $\frac{\partial E}{\partial w_{ij}^{(1)}}$ the last term which is derivative of $\frac{\partial net_j^{(1)}}{\partial w_{ij}^{(1)}} = f_i$ or simply i^{th} feature data is multiplied by $\delta_j^{(1)}$, hence; $\frac{\partial E}{\partial w_{ij}^{(1)}} = \delta_j^{(1)} f_i$. In the end, we have:

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = \begin{cases} \delta_j^{(1)}(h) f_i & if\ D<Q \\ \delta_j^{(1)}(b) f_i & if\ Q<D \end{cases} \tag{3.41}$$

When all partial derivatives are computed the network weights would be updated in the negative gradient direction. Introducing a constant γ as a learning rate or the step length of the correction, the corrections for the weights will be:

$$\Delta w_{ij}^{(1)} \begin{cases} -\gamma\delta_j^{(1)}(h) f_i & if\ D<Q \\ -\gamma\delta_j^{(1)}(b) f_i & if\ Q<D \end{cases} \quad \forall\, i=1,\ \ldots,\ N\ ; j=1,\ldots,\ M \tag{3.42}$$

$$\Delta w_{j1}^{(2)} \begin{cases} -\gamma\delta^{(2)}(h) o_j^{(1)} & if\ D<Q \\ -\gamma\delta^{(2)}(b) o_j^{(1)} & if\ Q<D \end{cases} \quad \forall\, j=1,\ldots,\ M \tag{3.43}$$

Usually there are more than one sample which are fed into the network. So if we have T pairs of samples like $\{(\boldsymbol{F}_1, D_1), (\boldsymbol{F}_2, D_2), \ldots, (\boldsymbol{F}_T, D_T)\}$ with N input features, the weight corrections are computed for each pattern and we get, for example, the corrections as $\Delta_1 w_{ij}^{(1)}, \Delta_2 w_{ij}^{(1)}, \ldots, \Delta_T w_{ij}^{(1)}$ for weight $w_{ij}^{(1)}$. Therefore the total amount of correction in the gradient direction is:

$$\Delta w_{ij}^{(1)} = \Delta_1 w_{ij}^{(1)} + \Delta_2 w_{ij}^{(1)} + \ldots + \Delta_T w_{ij}^{(1)} \tag{3.44}$$

To have previous calculations and then weight updates we can write them in matrix form and obtain necessary terms all together at the same time for all sample pairs. The matrices we have at the beginning are matrix F of size $T \times N$ of all input data, the matrix $W^{(1)}$ of size $N \times M$ of weights between the input layer and hidden layer, matrix $W^{(2)}$ of size $M \times 1$ of weights between hidden layer and single output unit.

In feed-forward step, the activation function in each node along with its corresponding derivative is calculated and stored in the right and left half of the node respectively. Then the transmitted values to the hidden layer is the matrix $net_{T\times M}^{(1)} = F \times W^{(1)}$, where the $(ij)^{th}$ element represents the transmitted inputs of sample i to hidden unit j. By computing the sigmoid of $net^{(1)}$ we get the hidden layer outputs of $O_{T\times M}^{(1)} = s\left(net^{(1)}\right)$ and its derivative which is $s'\left(net^{(1)}\right)_{T\times M} = O^{(1)} \odot \left(J_{T\times M} - O_{T\times M}^{(1)}\right)$ where $J_{T\times M}$ is an all-one matrix and the operator $\odot$ is an element-wise multiplication. Now the outputs of the hidden layer are mapped by second set of weights $W^{(2)}$ and enter into the output node. So we have $net_{T\times 1}^{(2)} = O^{(1)} \times W^{(2)}$, this is the vector of network output which is order quantities. The derivatives stored in this layer would be all-one matrix $J_{T\times 1}$ since the activation function is an addition function. By comparison of network output matrix $O_{T\times 1}^{(2)} = Q$ with demand matrix element by element we get the vector $\delta_{T\times 1}^{(2)}$ whose elements are either h or b if $D < Q$ or $Q < D$ respectively.

In order to obtain partial derivatives $\frac{\partial E}{\partial W^{(2)}}$ in a matrix, based on formulas we have

then:

$$\frac{\partial E}{\partial W^{(2)}} = \delta^{(2)} \odot O^{(1)} \tag{3.45}$$

and the correction is:

$$\Delta W^{(2)}_{M\times 1} = -\gamma (O^{(1)})^* \times \delta^{(2)} \tag{3.46}$$

Similarly, for $W^{(1)}$ correction we need first compute $\frac{\partial E}{\partial W^{(1)}}$. For this we have the matrix of backpropagation error $\delta^{(1)}_{T\times M} = \delta^{(2)}(W^{(2)})^* \odot s'(net^{(1)})$. Finally the matrix of partial derivatives $\frac{\partial E}{\partial W^{(1)}} = F^* \times \delta^{(1)}$ and correction is the matrix $\Delta W^{(1)}_{N\times M} = -\gamma F^* \times \delta^{(1)}$.

In a general form, let consider a network with L layers including input and output layers, so it has $L-2$ hidden layers. $W^{(l,\, l+1)}$ for $l = 1, \ldots, L-1$ denotes the weight matrix between layers l and $l+1$. Each layer except the input layer have a backpropagation error indicated by $\delta^{(l)}$ for $l = 2, \ldots, L$. Finally, $net^{(l)}$ and $O^{(l)}$ for $L = 1, \ldots, L-1$ define the input and output of each layer. We know that $O^{(1)} = F$ where F contains the input features. Starting from the output layer, the backpropagation error would be the vector $\delta^{(L)}$ whose elements are either h or b. The derivative of the last function w.r.t. the $W^{(L-1,L)}$ is:

$$\frac{\partial E}{\partial W^{(L-1,L)}} = \delta^{(L)} \odot O^{(L-1)} \tag{3.47}$$

and the correction is:

$$\Delta W^{(L-1,L)} = -\gamma (O^{(L)})^* \times \delta^{(L)} \tag{3.48}$$

For the hidden layers, the backpropafation errors are:

$$\delta^{(l)} = \delta^{(l+1)}(W^{(l,\, l+1)})^* s'(net^{(l)}) \quad for \;\; l = 2, \ldots, L-1 \tag{3.49}$$

Accordingly, the derivatives of the objective function w.r.t. the rest of the weights are calculated as follows:

$$\frac{\partial E}{\partial W^{(l,\, l+1)}} = (O^{(l)})^* \times \delta^{(l+1)} \quad for \;\; l = 1, \ldots, L-2 \tag{3.50}$$

and the corrections are:

$$\Delta W^{(l,\, l+1)} = -\gamma (O^{(l)})^* \times \delta^{(l+1)} \quad for \;\; l = 1, \ldots, L-2 \tag{3.51}$$

3.4 Method

Demand is assumed to come from different distributions. The distributions depend on some states which have two properties; first, follow a time-dependent model, second, are hidden. The model which produces the time series of states is a Markov model. Some probabilities serve as a veil between states and demand. The probability densities are defined by some other data as features besides of demand. Therefore, the joint probability distribution of demand and a state varies depending on some relations that explain the demand in each state.

Let D_t and $\boldsymbol{F}_t$ be the demand and a vector of features at time t, respectively. Moreover, s_t denotes the state at this time. In this part, a linear relation is considered between D_t and $\boldsymbol{F}_t$ like Equation (3.13). This relation is shown by the function $D_t = \psi\ (\boldsymbol{F}_t, W_{s_t})$ where W_{s_t} is a vector of linear coefficients corresponding to state s at time t. In order to have a more real data, an error has to be added to the deterministic part of the demand in this linear function:

$$D_t = \psi\ (\boldsymbol{F}_t, W_{s_t}) + \varepsilon_t \tag{3.52}$$

where ε_t has a normal distribution; hence $\varepsilon_t \sim \mathcal{N}(0, \sigma_{s_t})$. Where σ_{s_t} is a positive number and depends on the state at time t. Similarly, the conditional distribution of

demand given a state is as follows:

$$D_{t|s_t} \sim \mathcal{N}(\psi\left(\boldsymbol{F}_t, W_{s_t}\right), \sigma_{s_t}) \tag{3.53}$$

DNN is selected as a learning tool that maps the inputs to outputs in a way that an objective function would be optimized. In this problem, the Newsvendor cost function is the objective to be minimized. Regarding inputs of DNN, the available features are the main inputs at each time. Moreover, we need to introduce other inputs to capture the variations result from the distributions to which the demand belongs. These additional inputs are the unobservable states. However, the hidden states have direct effects on demand distribution. Thus, the most likely sequence of states should be obtained from the available data. To this aim, the LPHMM is applied to the features and demand.

First, the Baum-Welch algorithm converges and HMM parameters are estimated at final iteration using Equations (3.31), (3.32), and (3.33) on a training sample set. Afterward, the probabilities of the states for one step ahead of time are calculated by the Forward algorithm. This algorithm is based on the forward variable in Baum-Welch. The state sequence obtained by the Forward algorithm is not the most likely one, but rather produces the most likely state at each time step, given the previous data. The idea is similar to the Baum-Welch algorithm that benefits from conditional independence rules of HMM and gives the probabilities by a recursive method. Recalling the notation used in section 3.3.3, we have:

$$\alpha_t\left(s\right) = P\left(\mathcal{Y}_{1:\,t}, S_t = s\right) = \sum_{i=1}^{S} P\left(\mathcal{Y}_{1:\,t}, S_t = s, S_{t-1} = i\right) \tag{3.54}$$

Where $\mathcal{Y}_{1:\,t} = \{Y(1), \ldots, Y(t)\}$.

Using the chain rule and rewriting for $P\left(\mathcal{Y}_{1:\,t}, S_t = s, S_{t-1} = i\right)$, we then have:

$$\alpha_t\left(s\right) = \sum_{i=1}^{S} P(\mathcal{Y}_{1:\,t} \mid S_t = s, S_{t-1} = i, \mathcal{Y}_{1:\,t-1}) \\ P(S_t = s \mid S_{t-1} = i, \mathcal{Y}_{1:\,t-1}) P(S_{t-1} = i, \mathcal{Y}_{1:\,t-1}) \quad (3.55)$$

Since the last observation $Y(t)$ is conditionally independent of everything but S_t, and in Markov model, we know S_t only depends on S_{t-1}, Equation (3.55) could be written as:

$$\alpha_t\left(s\right) = P(\mathcal{Y}_{1:\,t} \mid S_t) \sum_{i=1}^{S} P(S_t = s \mid S_{t-1} = i) \alpha_{t-1}(i) \quad (3.56)$$

Finally, the probability of being in state j for a new time $t+1$ is:

$$P(S_{t+1} = j) = \sum_{i=1}^{S} \alpha_t(i) a_{ij} \quad (3.57)$$

where a_{ij} is the probability of transition from state i to state j and estimated recently.

Let $q_{t+1}(j)$ denote the probability obtained by Equation (3.57). The vector $\mathbf{q}_{t+1} = [q_{t+1}(j=1), \ldots, q_{t+1}(j=S)]$ contains the probabilities of all states at time $t+1$ and sums to one. If we round the maximum element to one and the rest to zeros, we get a vector $\mathbf{I}_{t+1}(s) = [0, \ldots, 1, \ldots, 0]$ which has the only non-zero element at s^{th} state. This means that the observation of $t+1$ is assigned to state $S_{t+1} = s$. Therefore, either vector $\mathbf{q}_{t+1}$ or $\mathbf{I}_{t+1}$ will be fed into a DNN as representative inputs for state information.

To compare the performance of DNN, the True model is considered as the best baseline in most of the figures. The Ordinary Least Square (OLS) model has been plotted along with the DNN and True model. These plots are in a ratio to True model. This lets the plots of all data sets be comparable. True model and OLS are explained before presenting the results.

3.4.1 True Model

For every data type, the cost of Newsvendor is calculated by the original parameters used to generate the data. Let $\tilde{\lambda} = \{A, \mathbf{W}, \sigma\}$ be the set of original parameters. The final Equation (3.57) of Forward algorithm gives the state probabilities for out of sample set using $\tilde{\lambda}$. These probabilities make the probability density function (pdf) for each time in two modes. First, state-mode (sm) which uses the rounded values of probabilities. The binary values are denoted recently by vector $\mathbf{I}$. Second, mixture-mode (mm) that takes the probabilities themselves by the vector $\mathbf{q}$. Before going to determine the order quantity, the pdf of demand should be specified at each time. Suppose that $\mathbf{I}_t$ and $\mathbf{q}_t$ are calculated for time t. Moreover, given the state, the pdf of demand at each time t is derived according to the relation (3.53). Let denote this relation in shorter form with $D_{t|s_t} = \mathcal{N}_{s_t}$. Thus, for each time t, a vector of all possible pdfs are as follows:

$$D_t \in \{\mathcal{N}_1, \ldots, \mathcal{N}_S\} \tag{3.58}$$

where S is the number of all states.
In the case of state-mode, pdf of demand is equivalent to the i_{th} normal distribution so that $\mathbf{I}(i) = 1$. Hence:

$$D_t^{sm} \sim \mathcal{N}_{\{s|\mathbf{I}_t(s)=1\}} \tag{3.59}$$

If Φ_t^{sm} denotes the cumulative distribution function of demand in this mode, the optimal order quantity is simply computed by Equation (3.2):

$$(Q_t^{sm})^* = (\Phi_t^{sm})^{-1}(\tau) \tag{3.60}$$

In mixture-mode, the calculations are less straightforward than previous mode due to the mixture Gaussian distribution at each time. Indeed, the set of all possible distributions are mixed proportional to the elements of the vector $\mathbf{q}$ at each time.

Distribution of demand at time t is a mixture Gaussian under vector $\mathbf{q}_t$ as below:

$$D_t^{mm} \sim \mathcal{N}_{\{\mathbf{q}_t\}} \tag{3.61}$$

Similar to state-mode the optimal order is:

$$(Q_t^{mm})^* = \Phi_t^{mm(-1)}(\tau) \tag{3.62}$$

$\Phi_t^{mm(-1)}(\tau)$ does not have a closed form to be evaluated at point τ directly. However, Φ_t^{mm} is a non-decreasing function, this value is found by a binary search. In final, two types of costs are computed for True model as follows:

$$C^{sm} = \sum_{t=1}^{T} C(Q_t^{sm}, D_t) \tag{3.63}$$

$$C^{mm} = \sum_{t=1}^{T} C(Q_t^{mm}, D_t) \tag{3.64}$$

where $C(Q, D)$ is the Newsvendor cost function as in Equation (3.1).

3.4.2 Ordinary Least Square

The ordinary least square is a regression model which relates some exploratory data to a response variable via a linear regressive equation. Here, demand is regressed on feature data. This model can be regarded as a Markov model with only one state. Let $\mathbf{F}_{T\times N}$ denote the matrix of features, $D_{T\times 1}$ the demand, and $\vec{\mathbf{w}}_{N\times 1}$ the vector of unknown linear coefficients. The linear multivariable regression has the following form:

$$D = \mathbf{F}\vec{\mathbf{w}} \tag{3.65}$$

The estimated vector of coefficients is given by:

$$\hat{\vec{\mathbf{w}}} = (\mathbf{F}^* R \mathbf{F})^{-1}(\mathbf{F}^* R) D \tag{3.66}$$

where R is a weight matrix which here is equally allocated to all samples. Then, for a set of new input variables of time t the mean of the output or response variable will be:

$$\hat{D}_t = \boldsymbol{F}_t \hat{\vec{\mathbf{w}}} \tag{3.67}$$

The errors made in this model are expressed in a vector $\mathbf{e} = D - \hat{D}$. $\mathbf{e}$ has a normal distribution, $\mathcal{N}(0, \sigma)$, where σ is the standard deviation of the errors. Thus, the demand at time t has the distribution written below:

$$D_t \sim \mathcal{N}(\boldsymbol{F}_t \hat{\vec{\mathbf{w}}}, \sigma) \tag{3.68}$$

Accordingly, the optimal order quantity using the formula (3.2) is:

$$(Q_t^{ols})^* = \Phi_t^{ols^{(-1)}}(\tau) \tag{3.69}$$

where Φ_t^{ols} is the cdf of demand at time t. The cost of this method is:

$$C^{ols} = \sum_{t=1}^{T} C(Q_t^{ols}, D_t) \tag{3.70}$$

3.5 Numerical Experiment

The proposed approach is applied to various synthetic data in this section. The performance of DNN is investigated by examination of the method on several data sets with different parameters. One of the issues of interest is to find out whether DNN could distinguish and model the state information without feeding the LPHMM outputs to DNN. Hence, the only inputs of DNN are the features in a set of models. These experiments and their results are presented in sections 3.5.2 and 3.5.3.

3.5.1 Data Generation

First, some independent data as features are needed to generate a time series of a data-driven hidden Markov. For this, some well-known distributions are used to have enough types of features that may exist in real data. Let name these independently distributed data as *raw* features. Table 3.2 shows the parameter selection and the associated distributions for *raw* features.

We can pick any number of features from any type of distributions indicated in Table 3.2. Suppose a required time series that has a length of T and N features recorded beside of demand at each time. Let $\mathbf{F}_{T\times N}$ denote the matrix of *raw* features. Each row of this matrix corresponds to each time and its columns contain intended feature types. Regarding value of T, it gets the integer numbers from $T \in \{25, 50, 100, 200, 400, 800\}$ for training sets and $T = 5000$, enough big set, for test. N is an arbitrary integer implies the number of features in different sample sets. In this experiment, $N \in \{1, 3, 5, 10\}$. If there exist n features of one type in a sample set, the parameters of the i^{th} feature for $i = 1, \ldots, n$ are computed as in Table 3.2. For example, let have $n = 2$ Normal distributed features, the parameters are $\mathcal{N}(10, 5)$ and $\mathcal{N}(20, 10)$ for $i = 1$ and $i = 2$, respectively. Let $\mathcal{N}_{i=1:2}$ show them together. This notation will be used to indicate the features in a data set in Table 3.3.

After producing matrices of *raw* features, a sequence of states generated from the Markov model is used to make time series demands. The transition probability matrix of Markov is the same for all the sample sets. It is arbitrarily chosen for a two-state Markov and equals to:

$$A = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

Table 3.2: Distributions and parameters of *raw* features

Distribution	Normal	Uniform	Exponential	Bernoulli
Parameters	$\mathcal{N}(10i, 5i)$	$\mathcal{U}(10i, 30i)$	$\exp(20i)$	$\text{Bern}(p = 0.75)$

Two more sets of parameters complete the demand observations;a set linear coefficients and the standard deviation of the error term added in each state as determined in relation (3.53). The former is chosen from an arbitrary coefficient matrix below:

$$\mathbf{W} = \begin{bmatrix} 0.490 & 0.230 & 0.166 & 0.120 & 0.140 & 0.0803 & 0.079 & 0.065 & 0.044 \\ 1.100 & 0.560 & 0.313 & 0.259 & 0.210 & 0.152 & 0.120 & 0.135 & 0.108 \end{bmatrix}$$

where the rows and columns correspond with the states and the features, respectively. Thus, a *raw* feature matrix $\mathbf{F}_{T\times N}$ should be multiplied by $\mathbf{W}_{2\times N}$ where the first N columns of $\mathbf{W}$ are picked. The values of $\mathbf{W}$'s elements have been chosen so that first, the effect of features in explaining the demand is approximately equal, and second, the value of demands is non-negative and reasonable in magnitude. In the case of a Bernoulli feature, its coefficients are 10 and 20 for the first and second states, respectively. An additional feature is considered in some sample sets which is not listed in Table 3.2 and it is a *irrelevant* feature. Consequently, its coefficients are zero for both states. Finally, the standard deviations for the error terms are $\sigma_{s_1} = N$ and $\sigma_{s_2} = 2N$, where N is the number of features in a sample set.

The sample data sets created for this part are presented in Table 3.3. We use the name of data sets briefly listed in this Table in the rest of the text.

3.5.2 Numerical Results

In this section, the numerical results are mostly presented in plots and figures. First, let present the True model results stated in section 3.4.1. Figure 3.4 shows the two costs of C^{sm} and C^{mm} relating to state-mode and mixture-mode, respectively. The cost of mixture-mode strictly outperforms the state-mode for all types of data. Thus, C^{mm} creates a baseline in most of the plots where the other approaches' costs are divided by C^{mm} to have comparable objective values. $b/h = 3$ indicates the ratio between Newsvendor parameters in the title of the legends of all plots. Moreover, unless it is mentioned, the plots are the average results over 50 sets simulated by the same parameters for each type of data to diminish the uncontrollable variations and

Table 3.3: Sample data sets

Name of Data Set	Features
`1n`	$\{\mathcal{N}_{i=1}\}$[a]
`1u`	$\{\mathcal{U}_{i=1}\}$
`1e`	$\{\exp_{i=1}\}$
`1n1u1e`	$\{\mathcal{N}_{i=1}, \mathcal{U}_{i=1}, \exp_{i=1}\}$
`4n1b`	$\{\mathcal{N}_{i=1:4}, \text{Bern}(p = 0.75)\}$
`4u1b`	$\{\mathcal{U}_{i=1:4}, \text{Bern}(p = 0.75)\}$
`4e1b`	$\{\exp_{i=1:4}, \text{Bern}(p = 0.75)\}$
`9n1ir`	$\{\mathcal{N}_{i=1:9}, irrelevant\}$
`9u1ir`	$\{\mathcal{U}_{i=1:9}, irrelevant\}$
`9e1ir`	$\{\exp_{i=1:9}, irrelevant\}$

[a] Parameters of i^{th} feature are computed as in Table 3.2

prevent overfitting or underfitting.

The estimated state sequence in binary or probability forms is the main output of LPHMM which explains the state information for DNN. Let Suppose that $\hat{\lambda} = \{\hat{A}, \hat{\mathbf{W}}, \hat{\sigma}\}$ is the Markov parameters that are estimated for all sample sets by LPHMM. The Baum-welch algorithm starts with a random initial model. A maximum number of iterations is the criteria to stop the algorithm. This number is set to 50 iterations. It is observed that this number is enough for some individual sets of any type in Figure 3.5(a).

We need to fix some parameters of DNN structure before training and feeding inputs. These parameters include:

- Number of input units: It depends on the number of features of each input matrix indicated by N.

- Number of output units: It is one as a single item Newsvendor is under study.

- Number of hidden layers: A DNN with two hidden layers is considered.

- Number of hidden (computing) units in each hidden layer: Let $M_1 = 50N$ and $M_2 = M1/2$ denote the number of hidden units in the first and the second

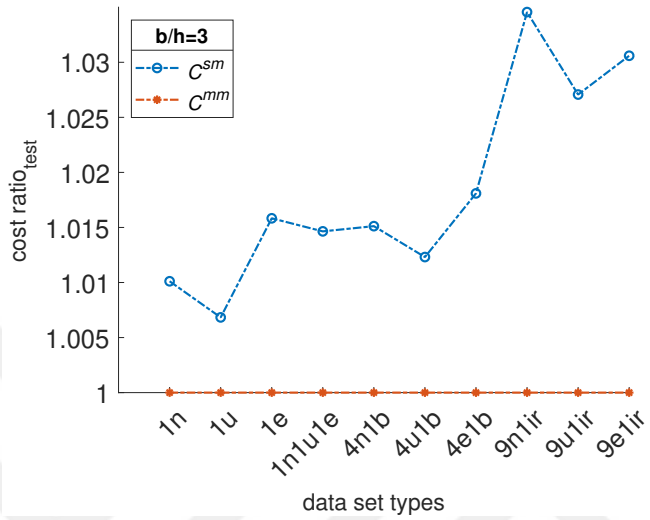


Figure 3.4: Cost ratio of state-mode and mixture-mode to the latter in True model for all data sets

hidden layer, respectively, where N is the number of inputs.

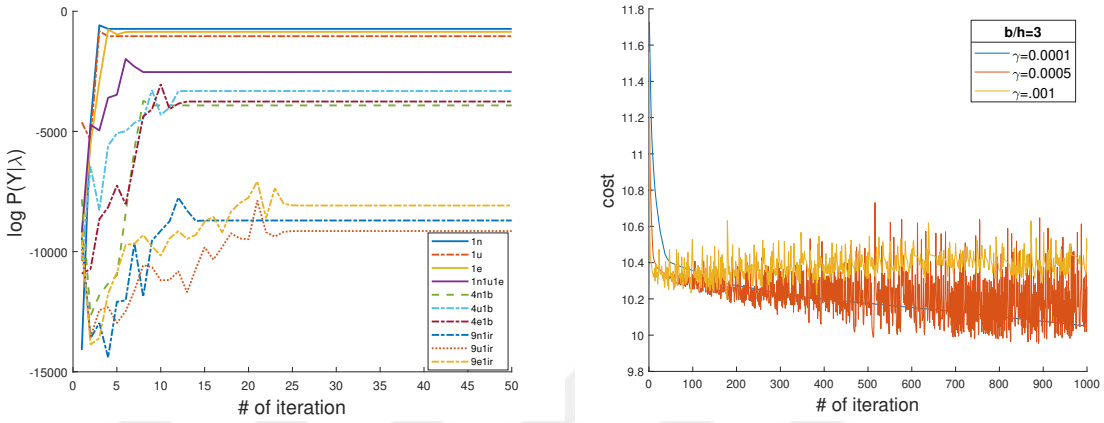
- Activation function: Sigmoid function is used as the activation function in computing units whose formula is:

$$s(x) = \frac{1}{1 + e^{-x}} \tag{3.71}$$

- Step size for updating weights: γ is in the set $\{0.0001, 0.0005, 0.001\}$ and the one resulting to the best cost function is selected.

- Initial weights: DNN starts with random weights.

- Stopping criteria: It is fixed on the maximum number of iterations of 1000. Figure 3.5(b) presents a convergence of DNN for an individual set of type for $\gamma \in \{0.0001, 0.0005, 0.001\}$.



(a) LPHMM convergence (b) DNN convergence of 1n for input set F

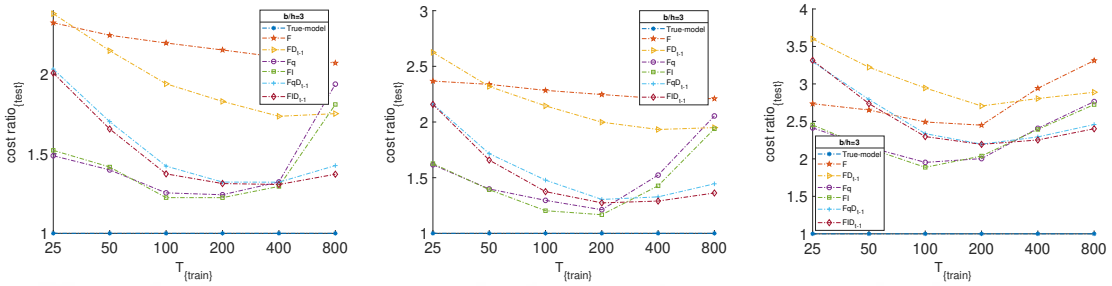
Figure 3.5: Convergence of algorithms for individual sets with $T = 200$

Various sets of inputs are fed into the DNN. This leads to relatively useful insights within the DNN efficiency. Many analyses are obtained from this diversification. Let recall the notations where $\mathbf{F}$ denotes the feature matrix, $\mathbf{I}$ the binary state information, $\mathbf{q}$ the probabilistic state information, and D_{t-1} the first lag of the demand. The different inputs are grouped in several sets and named in Table 3.4.

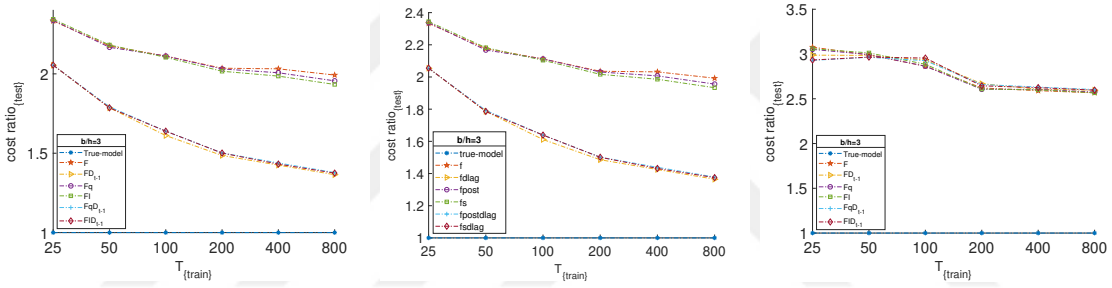
In Figure 3.6, the results of DNN for all data sets and inputs are presented. the X axis shows the length of training sets, while the Y axis includes the cost ratio for test sets. In all plots except Figure 3.6(f) and (i), the input set F is worse than other

Table 3.4: Different inputs for DNN

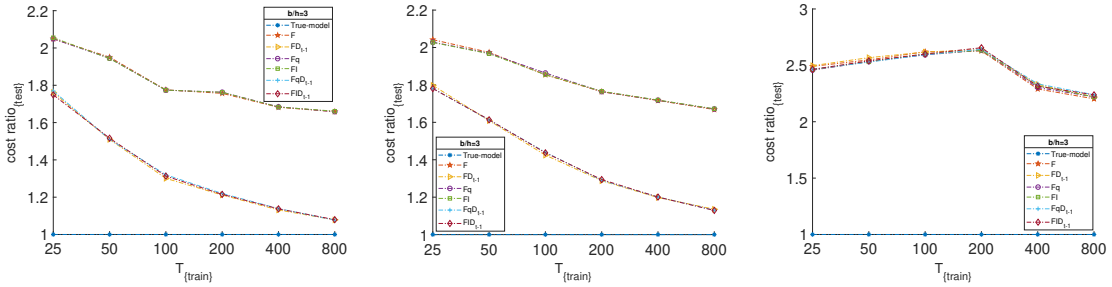
Input name	Input set
F	$\{\mathbf{F}\}$
FD_{t-1}	$\{\mathbf{F}, D_{t-1}\}$
Fq	$\{\mathbf{F}, \mathbf{q}\}$
FI	$\{\mathbf{F}, \mathbf{I}\}$
FqD_{t-1}	$\{\mathbf{F}, \mathbf{q}, D_{t-1}\}$
FID_{t-1}	$\{\mathbf{F}, \mathbf{I}, D_{t-1}\}$



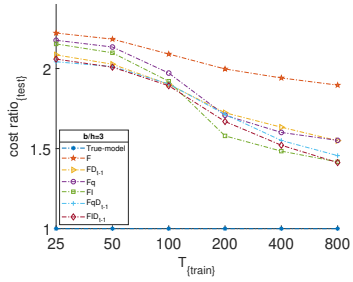
(a) 1n (b) 1u (c) 1e



(d) 4n1b (e) 4u1b (f) 4e1b

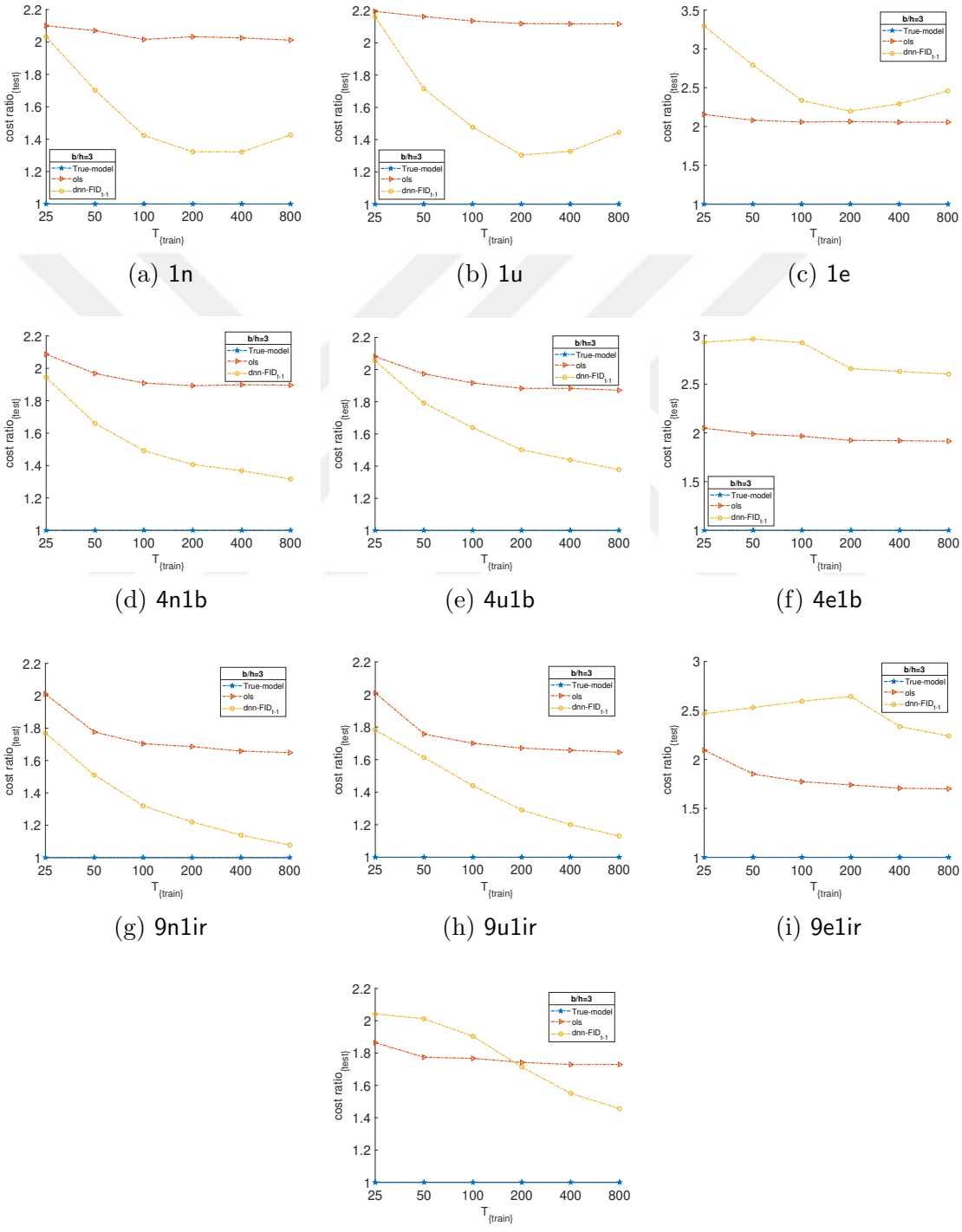


(g) 9n1ir (h) 9u1ir (i) 9e1ir



(j) 1n1u1e

Figure 3.6: Cost ratio of all data types in DNN



(a) 1n (b) 1u (c) 1e

(d) 4n1b (e) 4u1b (f) 4e1b

(g) 9n1ir (h) 9u1ir (i) 9e1ir

(j) 1n1u1e

Figure 3.7: Comparison of cost ratio for DNN and OLS

input sets. While in all figures, state information $\mathbf{I}$ with features outperforms the others. This happens with lagged demand in some figures. In Figure 3.7, then DNN model for input set $\mathsf{FID}_{\mathsf{t-1}}$ is plotted along with OLS model. It is noticeable that DNN could not outperform OLS where an exponential distributed feature exists in the input sets such as in Figures 3.7(c), (f), (i), and (j). This issue could be solved by normalizing the inputs. To improve DNN results, we need some analyses regarding *irrelevant* feature, DNN structure designing and DNN output sensitive analysis are implemented in section 3.5.3.

3.5.3 Some Analyses

In this section, some properties of DNN are investigated. These characteristics affect the outputs and help to figure out the mechanism followed by DNN. More knowledge about the algorithm eases the work more in analyzing real data and ensures we avoid adverse events like overfitting.

irrelevant Feature

In a real situation when relatively numerous features are selected for a model, some of them may be irrelevant to the response variable. Thus, an *irrelevant* feature is introduced to some data sets such as 9n1ir, 9u1ir, and 9e1ir. The speed of algorithms in recognizing this feature is of interest. To this end, the estimated coefficients in OLS, LPHMM, and weights exiting from this input in DNN are evaluated across the length of the training set.

Figure 3.8 represents the coefficient corresponding with this feature in OLS, LPHMM, and DNN. This value drops to zero for OLS and LPHMM as T increases. So, these models could discern the feature to be ineffective. However, the plot for DNN is indifferent to T and cannot show this issue. DNN may not transfer the information of *irrelevant* feature to the output unit. Therefore, since DNN is a nonlinear and complex network, we did a sensitivity analysis for the relation of changes in *irrelevant* feature and the resulted changes in output. It can be seen that DNN distinguishes

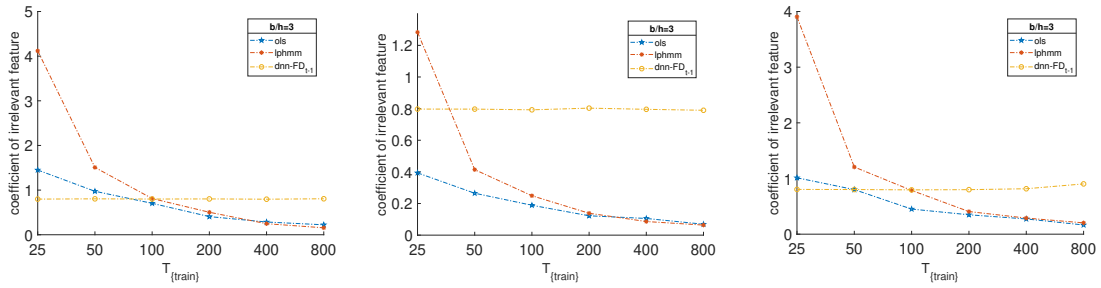
this feature automatically in Figure 3.9.

Different DNN Structure

In Table 3.5, several DNN structures are designed based on these two RoTs. Let $\mathcal{M}$ denote the total number of hidden units. The main difference is the proportional allocation of $\mathcal{M}$ to the hidden layers.

A new data set is generated to which the various DNNs be applied with normalized inputs. This set is named `2stdn1c` that is made up of two standard normal and one as a constant as features. The coefficients of the two states and the standard deviation of the error terms are as follows:

$$\mathbf{W} = \begin{bmatrix} 10 & 0 & 100 \\ 0 & 10 & 100 \end{bmatrix} \qquad \sigma = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$



(a) `9n1ir` (b) `9u1ir` (c) `9e1ir`

Figure 3.8: Coefficient of *irrelevant* feature

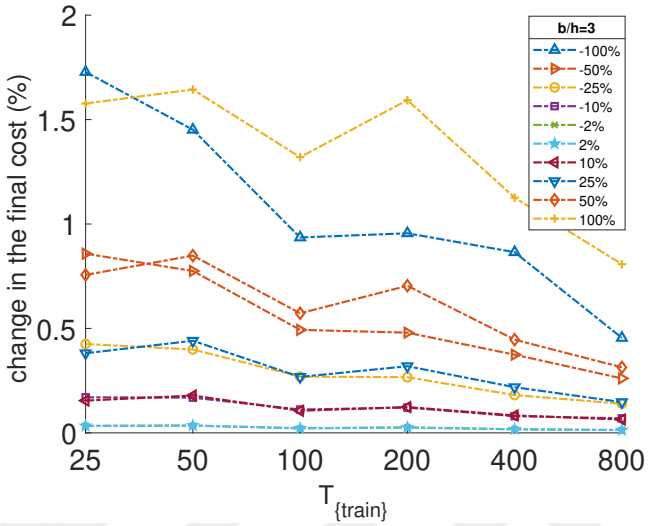


Figure 3.9: Sensitive analysis for *irrelevant* feature for 9n1ir and input set FD_{t-1}; Changes are depicted in absolute value

Table 3.5: Different structures of DNN

Number of hidden layers	Name of the structure	$\mathcal{M}$	Number of hidden units in each hidden layer
1	$\mathsf{STR}_{1,\mathsf{KeLiu}}$	$N+\sqrt{T}$	$M_1 = N+\sqrt{T}$
	$\mathsf{STR}_{1,5}$		$M_1 = 5$
	$\mathsf{STR}_{1,10}$		$M_1 = 10$
	$\mathsf{STR}_{1,20}$		$M_1 = 20$
	$\mathsf{STR}_{1,50}$		$M_1 = 50$
2	$\mathsf{STR}_{2,\mathsf{KeLiu}}$	$N+\sqrt{T}$	$M_1 = M_2 = (N+\sqrt{T})/2$
	$\mathsf{STR}_{2,\mathsf{Huang}}$	$2\sqrt{(m+2)T}$	$M_1 = \sqrt{(m+2)T + 2\sqrt{T/(m+2)}}$ $M_2 = m\sqrt{T/(m+2)}$
	$\mathsf{STR}_{2,1}$		$M_1 = M_2$
	$\mathsf{STR}_{2,2}$		$M_1 = 2M_2$
	$\mathsf{STR}_{2,3}$		$2M_1 = M_2$
3	$\mathsf{STR}_{3,\mathsf{KeLiu}}$	$N+\sqrt{T}$	$M_1 = M_2 = M_3 = (N+\sqrt{T})/3$
	$\mathsf{STR}_{3,1}$		$M_1 = 1.5M_2 = 3M_3$
	$\mathsf{STR}_{3,2}$		$3M_1 = 1.5M_2 = M_3$
4	$\mathsf{STR}_{4,\mathsf{KeLiu}}$	$N+\sqrt{T}$	$M_1 = M_2 = M_3 = M4 = (N+\sqrt{T})/3$

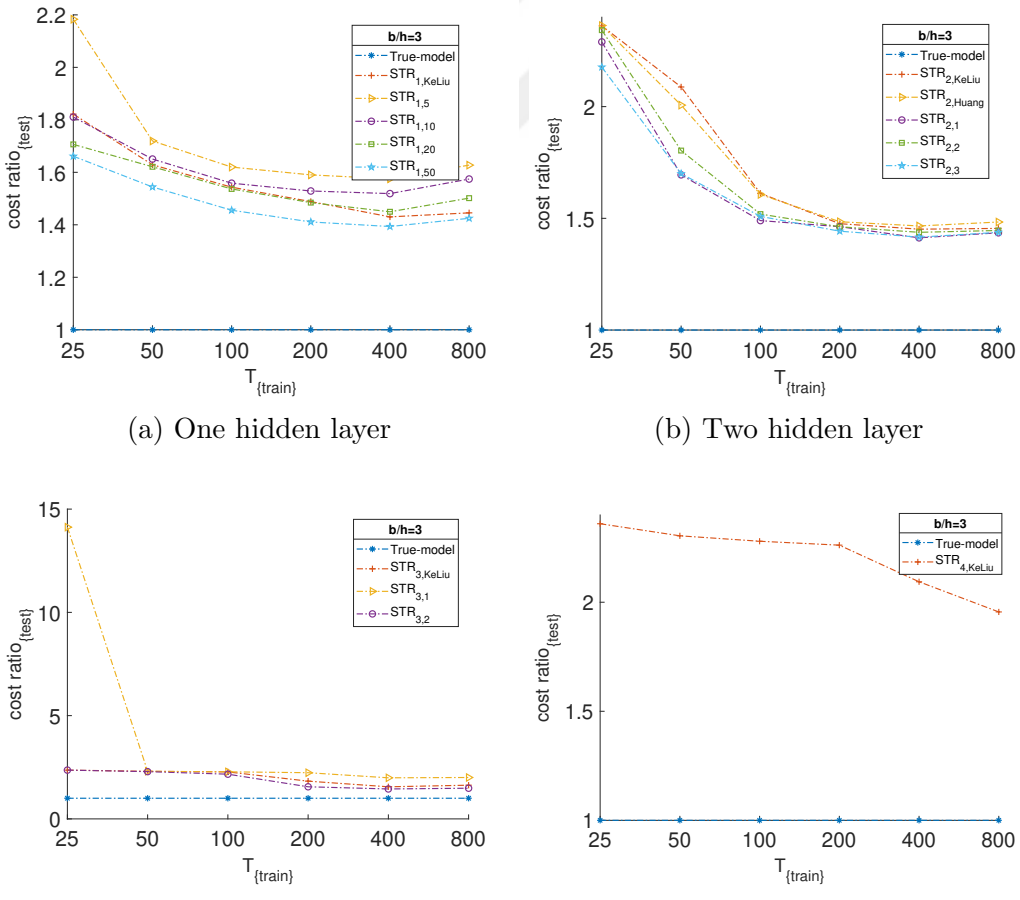
If we denote the standard normal distribution by $\mathcal{Z}$, the demand is as below for each state:

$$D_{s_1} = 10\mathcal{Z}_1 + 100 + \varepsilon_1 \tag{3.72}$$

$$D_{s_2} = 10\mathcal{Z}_2 + 100 + \varepsilon_2 \tag{3.73}$$

Where $\varepsilon_i \sim \mathcal{N}(0, 2)$ for $i = 1, 2$. The results for this data set are plotted based on average costs over all input sets listed in Table 3.4.

Figure 3.10 shows the results for all structures. Plot 3.10(a) demonstrates that in



(a) One hidden layer

(b) Two hidden layer

(c) Three hidden layer

(d) Four hidden layer

Figure 3.10: Cost of different DNN structures for data set `2stdn1c`

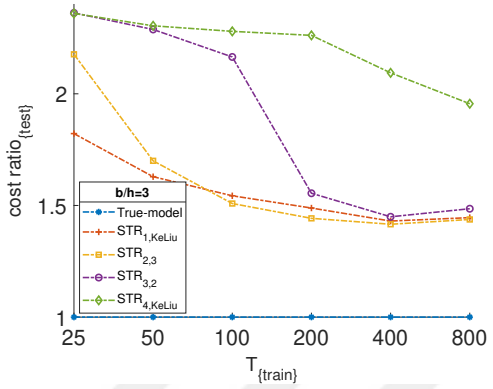
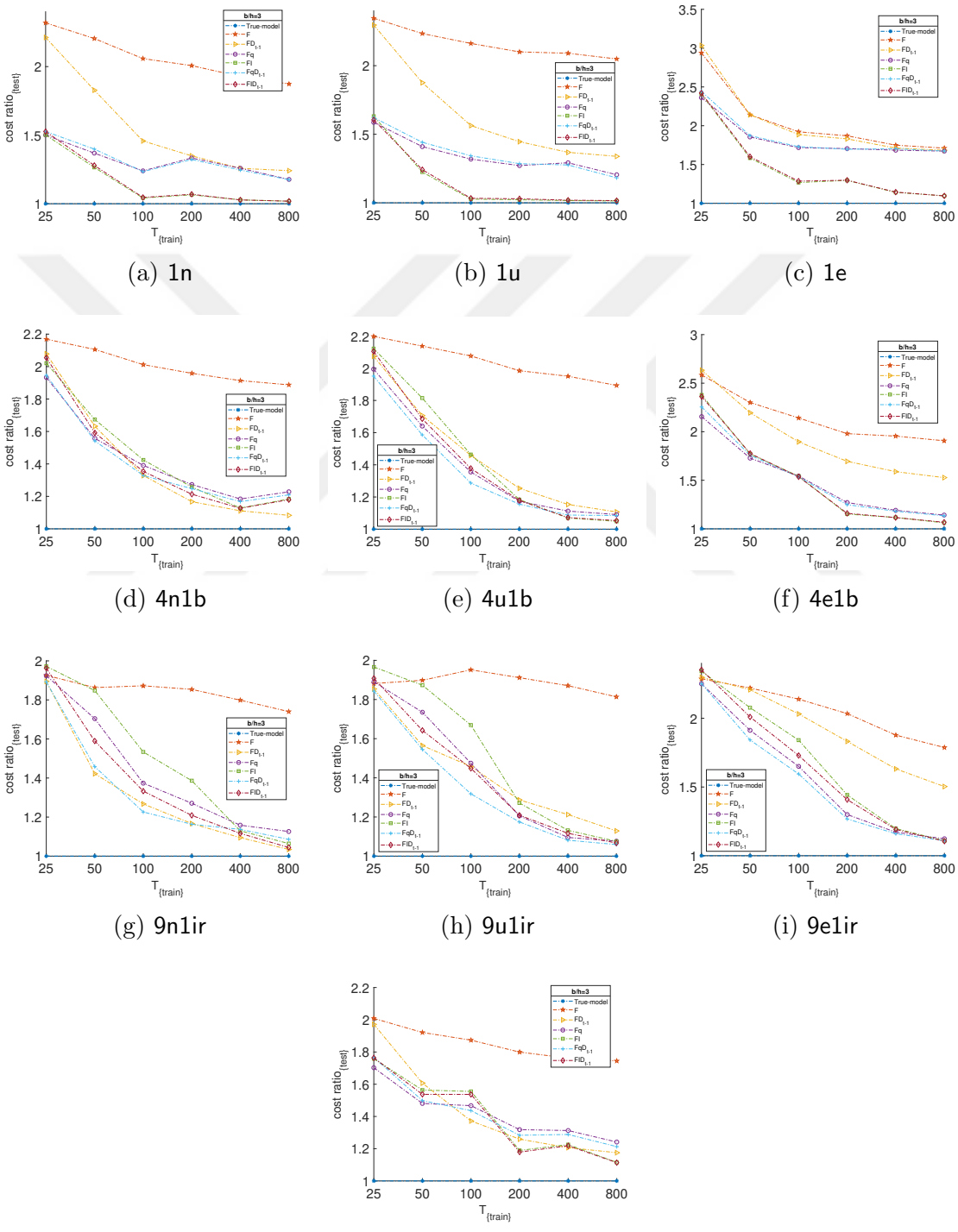


Figure 3.11: Comparison of the best of structures for data set 2stdn1c

a one hidden layer DNN, the number of units specified by *KeLiu* finally meets the others at $T = 800$. Due to the time efficiency, the *KeLiu* rule is more reasonable for one hidden layer DNN. The plots are close in each of two and three hidden layer DNN structures. However, the last structure is picked for them to compare. Only one structure is considered in the case of four hidden layers DNN.

In general, the best structure is selected from each plot of Figure 3.10. They are depicted along with each other to find the best structure in Figure 3.11. It could be seen that for all length of T except $T = 100$ and $T = 200$ the structure $\mathsf{STR}_{2,3}$ from a two hidden layer DNN is strictly lower than the $\mathsf{STR}_{1,\mathsf{KeLiu}}$, $\mathsf{STR}_{3,2}$, and $\mathsf{STR}_{4,\mathsf{KeLiu}}$ of one, two, and four hidden layer DNN, respectively.

The optimal DNN structure which is $\mathsf{STR}_{2,3}$ is applied to all normalized data sets. In Figure 3.12, the obvious result is that the features F lonely are not sufficient for DNN to capture the HMM. However, when we introduce state information either in the binary or probability form, the DNN reaches an acceptable level of cost ratio in all plots. On the other hand, DNN may figure out the HMM when the first lagged of demand is fed into the network for some data sets but not always. Two other data sets are generated so that the lagged demand is important in one of them and is ineffective in the other. This is to investigate the effect of demand on the results. Let consider a standard normal and one constant as features. The following sets are



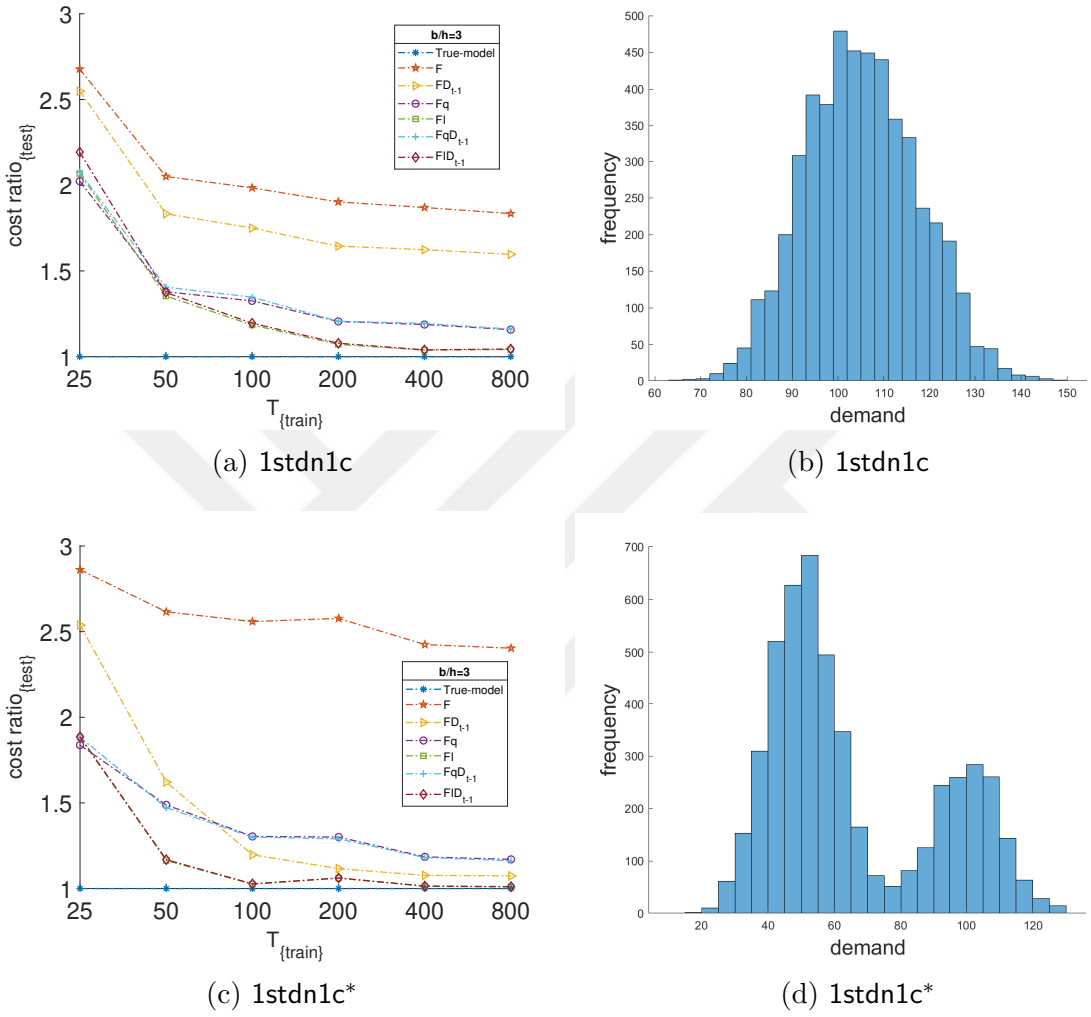
(a) 1n (b) 1u (c) 1e

(d) 4n1b (e) 4u1b (f) 4e1b

(g) 9n1ir (h) 9u1ir (i) 9e1ir

(j) 1n1u1e

Figure 3.12: Cost ratio for normalized data by the $\mathsf{STR}_{2,3}$ structure



(a) 1stdn1c (b) 1stdn1c

(c) 1stdn1c* (d) 1stdn1c*

Figure 3.13: Cost ratio and the demand distribution to investigate the demand effect as an input in DNN

created:

$$\text{1stdn1c} \quad D_{s_1} = 10\mathcal{Z} + 100 + \varepsilon_1 \tag{3.74}$$

$$D_{s_2} = 10\mathcal{Z} + 115 + \varepsilon_2 \tag{3.75}$$

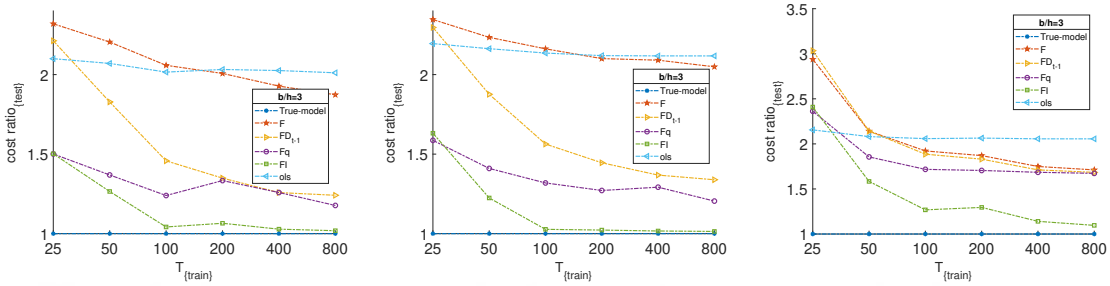
$$\text{1stdn1c}^* \quad D_{s_1} = 10\mathcal{Z} + 50 + \varepsilon_1 \tag{3.76}$$

$$D_{s_2} = 10\mathcal{Z} + 100 + \varepsilon_2 \tag{3.77}$$

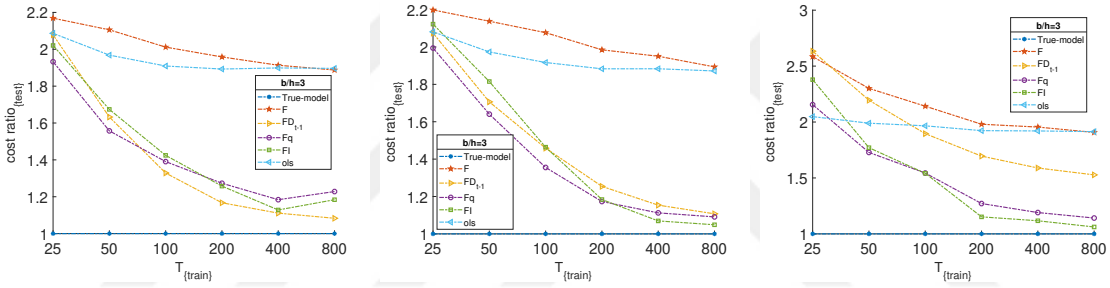
Where $\varepsilon_i \sim \mathcal{N}(0, 2)$ for $i = 1, 2$. The DNN costs are depicted along with the demand distribution for test set in Figure 3.13.

According to the figures for data set `1stdn1c`*, it can be seen that the DNN results in a low cost with lagged demand. By looking at its demand histogram which shows a separable mixture normal distribution we can conclude that the network keeps the state similar to the last time. This means when a level of demand is revealed, the next demand is at that level almost surely. This issue is investigated to see whether the two above examples prove the claimed relations in all data sets. In Figure 3.14 and corresponding demand histograms in Figure 3.15, it could be seen that when the demand has a multi-modal histogram, the lagged demand as a feature would be important in explaining the next time demand. In this Figure, other unuseful plots of Figure 3.12 are removed.

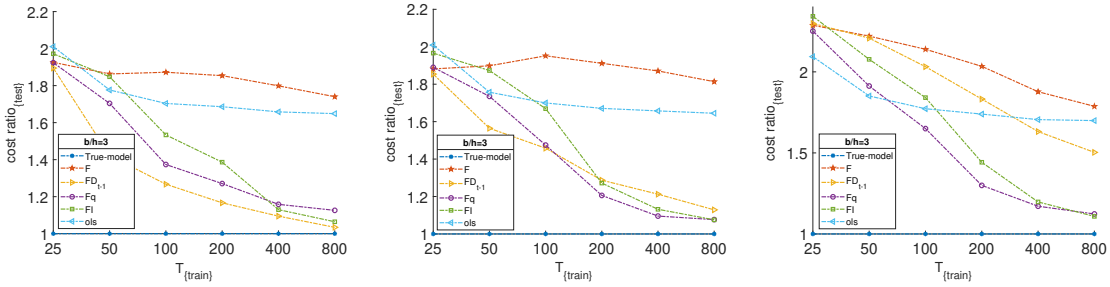
It is proved that unimodality causes DNN to work better with state information than lagged demand and even *ols* method.



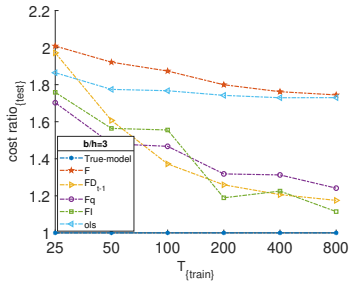
(a) 1n (b) 1u (c) 1e



(d) 4n1b (e) 4u1b (f) 4e1b

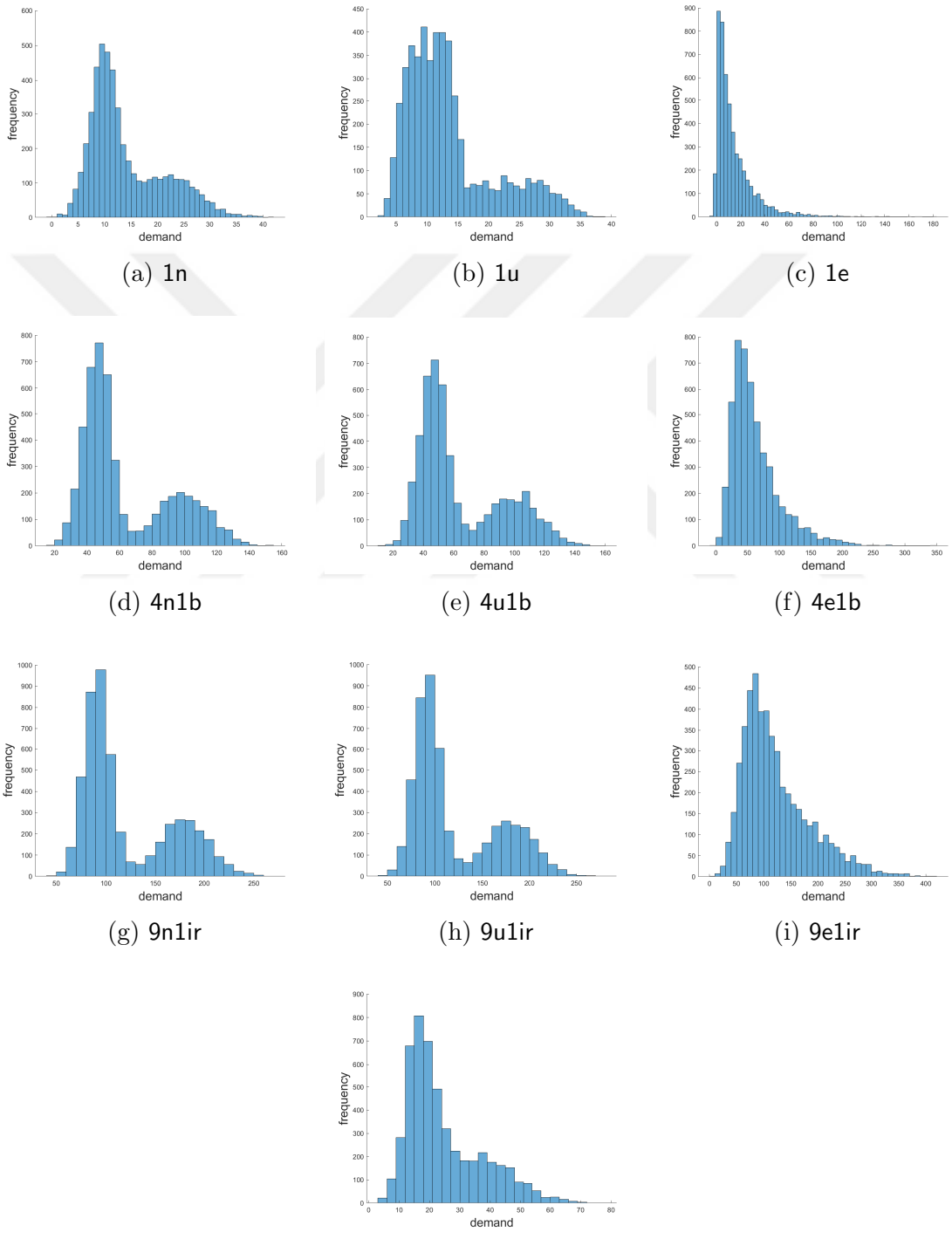


(g) 9n1ir (h) 9u1ir (i) 9e1ir



(j) 1n1u1e

Figure 3.14: Cost ratio for normalized data along with ols



(a) 1n (b) 1u (c) 1e

(d) 4n1b (e) 4u1b (f) 4e1b

(g) 9n1ir (h) 9u1ir (i) 9e1ir

(j) 1n1u1e

Figure 3.15: Demand frequency in test sets

3.6 Non-linear HMMNV

According to the results, it is concluded that state information is crucial in the presence of HMM. This importance would be more appealing when no obvious states are observed. Thus, a time series might not reflect the HMM in itself apparently, but the states are hidden in coefficients explaining the demand. This situation is more likely to occur where the demand has a nonlinear relationship with features. Essentially, this means that demand is regressed on features in a specific nonlinear form at each state generated by the Markov model. The LPHMM, described in section 3.3, does not work for nonlinear relations anymore. Therefore, a method is required to capture them efficiently so that the most probable sequence of states is obtained. Given this new problem, designing a DNN for the stated purpose is of interest. We consider DNN since it is one of the machine learning methodologies that can model highly non-linear functions and its training can be performed efficiently using gradient methods.

3.6.1 Demand Data and Notation

The historical data of the problem can be represented using tuples of feature vectors and demand realizations as

$$\mathcal{D} = \{(\boldsymbol{F}_1, D_1), (\boldsymbol{F}_2, D_2), \ldots, (\boldsymbol{F}_T, D_T)\}, \tag{3.78}$$

where T denotes the number of periods. In Equation (3.78), the vector of $\boldsymbol{F}_t$ for each period of $t = 1, 2, \ldots, T$ consists of N different features $f_{1t}, f_{2t}, \ldots, f_{Nt}$. Table 3.6 describes all the variables that are used in this study.

Table 3.6: Description of the variables of the model.

Variable	Description
D_t	demand at time t
Q_t	order quantity at time t
B_i	base demand in state i
f_{nt}	n^{th} feature observed at time t
$\mathbf{F}_t$	vector of independent and identically distributed of N features at time t
T	number of periods or sample observations
β_n	linear coefficient of n^{th} feature with demand
S_t	state of the Markov chain at time t
$\mathbf{q}_t$	vector of states probability
ψ^E	emission network function which maps the features $\mathbf{F}_t$ to D_t partially
$\mathbf{W}^E$	set of parameters of emission network function ψ^E
ψ^{NV}	newsvendor network function which maps $\mathbf{F}_t$ to Q_t partially
$\mathbf{W}^{NV}$	set of parameters of the newsvendor network function ψ^{NV}
ϵ_t	error term as the nonsystematic part of the demand
$\mathcal{N}(\mu, \sigma)$	Gaussian distribution with mean μ and standard deviation σ
Λ	likelihood function of the hidden Markov model
a_{ij}	probability of transition from state i to state j
A	transition probability matrix of the Markov model
E	the vector of the probability density functions of the hidden states
π	initial probability of the hidden Markov chain
α_t	forward parameter of the Baum-Welch algorithm
γ_1	learning rate of updating $\mathbf{W}^E$
γ_2	learning rate of updating $\mathbf{W}^{NV}$
γ_3	learning rate of updating A
η	the coefficient of the trade-off between newsvendor cost and the Likelihood

3.6.2 Model

We consider the general form of the function (3.11) and substitute it in the Equation (3.10) as suggested by Neghab et al. (2021b) to get

$$D_t = \sum_{i=1}^{S} I(S_t = i)B_i + \psi(f_{1t}, f_{2t}, \ldots, f_{Nt}) + \epsilon_t, \tag{3.79}$$

where $I(x) = 1$ if x is true and 0 otherwise. This implies a linear relation between the Markov chain-dependent and the features-dependent parts of the demand in each state and unknown complex relation between D_t and $\mathbf{F}_t$. More specifically, we assume that the unknown set of parameters of the function of features that constitutes the demand partially is $\mathbf{W}$. We then assume the following form

$$D_t = \sum_{i=1}^{S} I(S_t = i)B_i + \psi(\mathbf{F}_t, \mathbf{W}) + \epsilon_t. \tag{3.80}$$

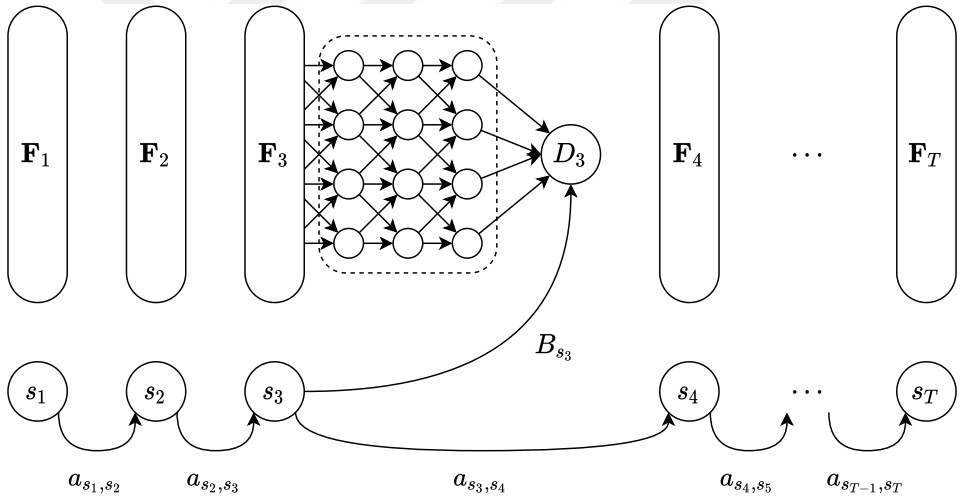


Figure 3.16: The effects of the features and the evolution of the state on the demand.

Figure 3.16 shows how demand is generated by features and states: D_3 is a function of the observed features F_3 and the unobservable state s_3.

The inventory optimization problem can be written as:

$$\min_{\mathbf{Q}} \; NVC(Q) = E_D\left[h(Q-D)^+ + b(D-Q)^+ | D_1, D_2, \ldots, D_{t-1}, \mathbf{F}_{t-1}\right]. \tag{3.81}$$

We propose to fit a joint estimation and optimization model to sample data and test the performance of the model out of sample with the objective of minimizing

the cost in the out of sample data. Minimizing the cost function requires deriving an accurate belief about the state of the Markov chain at each point in time and solving the inventory optimization problem at that time. These two problems can be stated as:

(1) Hidden Markov Model Problem:

First, we consider the problem of maximizing the likelihood of the observed sequence of demands that undergoes a hidden Markov model. Given a set of demand observations $D = D_{1:T}$, HMM with finite hidden state space S consists of a triple model parameters $\lambda(\pi, A, E)$, where $\pi = \{\pi_i = P(S_1 = i)\}$ is the prior probabilities of s_i being the first state of the demand observations. $A = \{a_{ij}\}$ is the state transition probabilities matrix. $E = \{p_1, p_2, \ldots, p_S\}$ is the probability density functions of the hidden states where $p_j(D_t) = P(D_t|S_t = j)$. The optimal solution of the following formulation is the model parameters set λ that is most likely to generate the observed demand sequence

$$\begin{aligned} \max_{\pi,A,E,} \quad & \Lambda = \log[P(D|\lambda)] \\ \text{s.t.}: \quad & A \cdot 1 = 1 \\ & \pi \cdot 1 = 1, \end{aligned} \tag{P-HMM}$$

where the first constraint characterizes the transition probability from hidden state S_{t+1} into S_t , and $\sum_j \{a_{ij}\} = 1$, and the second one satisfies the relation between observation D_t and hidden state S_t at time t, and $\sum_j \{p_j(D_t)\} = 1$.

(2) Newsvendor Problem:

The second problem is finding the network that sets the best order quantity given the state sequence $S = S_{1:T}$ estimated from the first problem that has most probably

generated the demand sequence

$$\begin{aligned} \min_{Q\geq 0} \quad & NVC(Q|S) \\ \text{s.t.}: \quad & Q_t = B_i + \psi^{NV}(\mathbf{F}_t, \mathbf{W}^{NV}) \quad ;\forall t = 1,\ldots,T,\ i \in S \\ & \mathbf{W}^{NV} \in \mathcal{R}. \end{aligned} \tag{P-NV}$$

However, these two objectives do not always align necessarily. We use a linear scalarization method to formulate these two problems as a single-objective optimization

$$\max_{Q\geq 0,\mathbf{q}} \ \text{Loss} = -\eta\Lambda + (1-\eta)NVC, \tag{P-HMMNV}$$

where $\mathbf{q}$ is the probability of the states effective at the time of making decision on order quantity Q.

To solve the above problem, we consider deep learning as it is one of the machine learning methodologies that can model both highly non-linear functions and the Markov chain using historical data and its training can be performed efficiently using gradient methods.

3.6.3 Recurrent Network for Solving Markov Modulated and Data-Driven Newsvendor

In this study, we propose a new algorithm referred to as HMMNV that utilizes the machine learning method of Deep Neural Networks (DNN) for solving the proposed model. We unify the objective functions by integrating of Markov chain and the newsvendor problem in a network. To this end, we propose a two-head neural network in which these objective functions are optimized simultaneously. The suggested network comprises two networks of HMM and the newsvendor. The most likely sequence of states is obtained from the available data by the HMM network. The state information, $\mathbf{q}$, that influences the order quantity partially along with the available features as the inputs of the newsvendor network completes the order quantity at each

time period. Figure 3.17 shows this integration and represents the folded version of the expanded recurrent network over time.

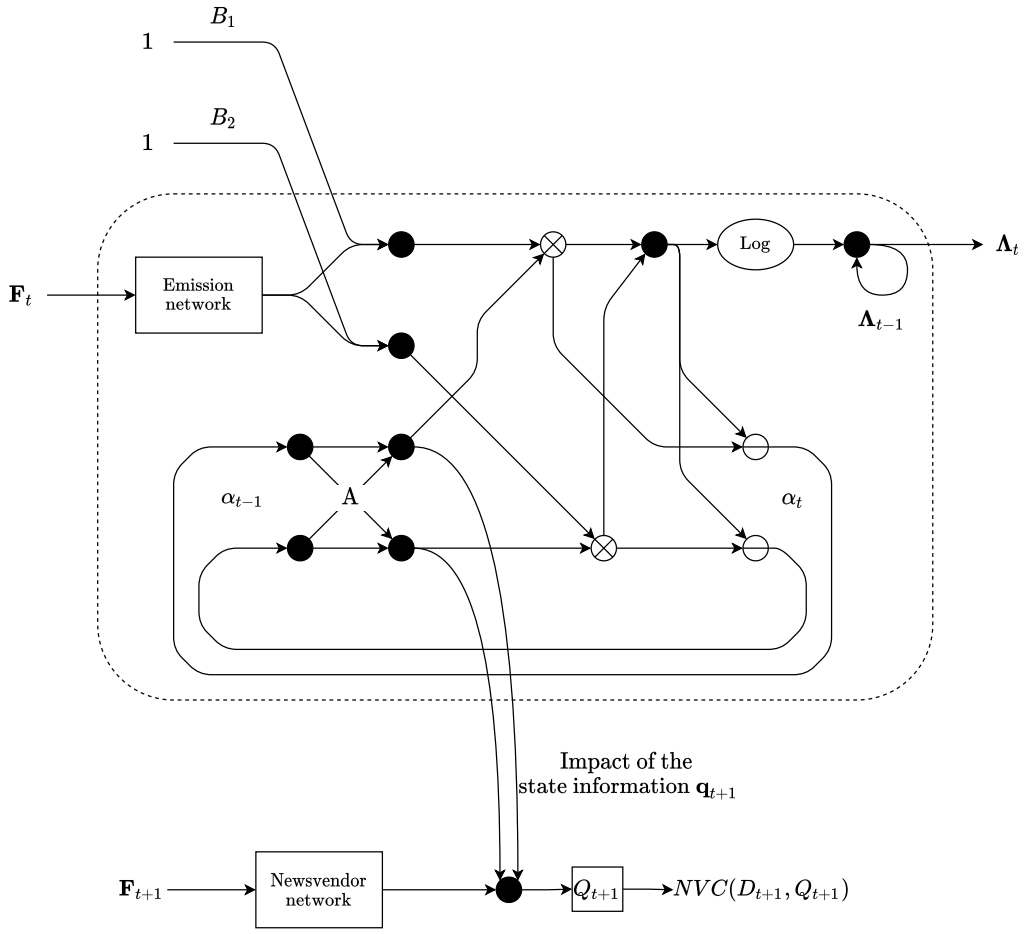


Figure 3.17: HMMNV network. This figure includes three neural networks: 1- Hidden Markov Model (HMM) network (dashed rectangle), 2- Newsvendor network, and 3- Emission network. The emission network is embedded into HMM network. The network weights denoted by B and A are the base demand associated with each state and the transition probabilities between states, respectively. All other connections have weight=1. Filled units (circles) denote summation, crossed units multiply their inputs, and units divided by a line, divide one input by the other. In fact, this network is repeated in the number of observations so that constructs a network over time (recurrent network).

In addition to these two networks, we estimate the base demand B and the function ψ in equation (3.80) by a separate neural network in each state. We refer to this network as the emission network whose outputs are used as likelihoods of an

observation given some model parameters. The emission network is embedded into the HMM network.

To estimate the hidden Markov model and train its network we can unfold the recurrent network and treat it as a feed forward network. Feeding and backpropagation steps of a neural network estimation is equivalent to the forward and backward steps of the well-known Baum-Welch algorithm which is used to estimate HMM. The likelihood of each demand observation is estimated by the probability density functions based on a normal distribution as

$$p_i(D_t) \sim \mathcal{N}\left(\mu_i + \psi^E(\mathbf{F}_t, \mathbf{W}^E), \sqrt{\sigma_i^2 + \sigma_\epsilon^2}\right), \tag{3.82}$$

where, μ_i is the mean of the base demand in state i. In the forward step, a forward term is defined at each time for each state of the model as

$$\alpha_t(s) = P(D_{1:t}, S_t = s) = P(D_{1:t}, S_t = s, S_{t-1} = i), \;\; D_{1:t} = \{D_1, D_2, \cdots, D_t\}, \tag{3.83}$$

where $D_{1:t} = \{D_1, D_2, \cdots, D_t\}$. Using the chain rule and rewriting for $P(D_{1:t}, S_t = s, S_{t-1} = i)$, we then have

$$\begin{aligned} \alpha_t(s) = \sum_{i=1}^{S} & P(D_{1:t}|S_t = s, S_{t-1} = i, D_{1:t-1}) \\ & P(S_t = s|S_{t-1} = i, D_{1:t-1}) P(S_{t-1} = i, D_{1:t-1}). \end{aligned} \tag{3.84}$$

Since the last observation D_t is conditionally independent of everything but S_t, and in the Markov model, we know that S_t only depends on S_{t-1}, Equation (3.84) could be written as

$$\alpha_t(s) = P(D_{1:t}|S_t = s) \sum_{i=1}^{S} P(S_t = s|S_{t-1} = i) \alpha_{t-1}(i). \tag{3.85}$$

Finally, the probability of being in state j for a new time $t+1$ is

$$P(S_{t+1} = j) = \sum_{i=1}^{S} \alpha_t(i) a_{ij}, \tag{3.86}$$

where a_{ij} is the most recent estimation of the probability of transition from state i to state j. Let $q_{t+1}(j)$ denote the probability obtained by Equation (3.86). The vector $\mathbf{q}_{t+1} = [q_{t+1}(j=1), \ldots, q_{t+1}(j=S)]$ contains the probabilities of all states at time $t+1$ which sum to one. Vector $\mathbf{q}_{t+1}$, as the state information, is then multiplied by some network weights and builds the base demand which is represented as the connection between HMM and newsvendor networks in Figure 3.17. The sum of the values obtained from state-dependent and feature-dependent is the estimation for the order quantity.

In the estimation process, scaling is required in the implementation of the forward-backward algorithm. Let us consider the forward term of Equation (3.83) in the re-estimation procedure and write as

$$\begin{aligned} \alpha_t(s) &= P_\lambda(D_0, \cdots, D_t, S_t = s) \\ &= \sum_{s_1 s_2 \ldots s_{t-1}} P_\lambda(D_0, \cdots, D_t, S_t = s \,|s_1 s_2 \ldots s_{t-1}) P_\lambda(s_1 s_2 \ldots s_{t-1}) \\ &= \sum_{s_1 s_2 \ldots s_{t-1}} \left[\prod_{i=1}^{t} p_{s_i}(D_i) \prod_{i=1}^{t-1} a_{s_i s_{i+1}} \right] \end{aligned} \tag{3.87}$$

Scaling the Terms:

All the involving terms in Equation (3.87) are on a probability scale meaning that they are less than one. Therefore, the summation rapidly drops to zero with an exponential rate. The result is too small and may exceed the machine precision and relative errors round to zero in floating-point. To solve this problem, a scaling is proposed and used by Levinson et al. (1983) to keep all $\alpha_t(s)$'s bounded at each induction step. This scaling factor only depends on time t and not the current state s. Corresponding computations include two parts:

- Initialization

$$\begin{aligned} \ddot{\alpha}_1(i) &= \alpha_1(i), \\ c_1 &= \frac{1}{\sum_{i=1}^{S} \ddot{\alpha}_1(i)}, \\ \hat{\alpha}_1(i) &= c_1 \ddot{\alpha}_1(i) \end{aligned} \tag{3.88}$$

- Induction

$$\begin{aligned} \ddot{\alpha}_1(i) &= \sum_{j=1}^{S} \hat{\alpha}_{t-1} a_{ji} p_i(D_t), \\ c_t &= \frac{1}{\sum_{i=1}^{S} \ddot{\alpha}_t(i)}, \\ \hat{\alpha}_t(i) &= c_t \ddot{\alpha}_t(i) \end{aligned} \tag{3.89}$$

The coefficient c_t at each step depends on t. $\hat{\alpha}_t(i)$ is the modified forward variable which sums to one, $\sum_{i=1}^{S} \hat{\alpha}_t(i) = 1$. It is easy to see that

$$\hat{\alpha}_t(i) = \left(\prod_{\tau=1}^{t} c_\tau \right) \alpha_t(i). \tag{3.90}$$

By using this new forward algorithm, obtained in the last step, we have

$$\begin{aligned} 1 &= \sum_{i=1}^{S} \hat{\alpha}_T(i) = \sum_{i=1}^{S} \left(\prod_{t=1}^{T} c_t \right) \alpha_T(i) \\ &= \left(\prod_{t=1}^{T} c_t \right) \sum_{i=1}^{S} \alpha_T(i) = \left(\prod_{t=1}^{T} c_t \right) P(D|\lambda). \end{aligned} \tag{3.91}$$

Let $\mathbf{C} = \prod_{\tau=1}^{t} c_\tau$, then $P(D|\lambda) = 1/\mathbf{C}_T$. The logarithmic form of the likelihood function is then

$$\Lambda = \log[P(D|\lambda)] = -\sum_{t=1}^{T} \log c_t. \tag{3.92}$$

In the next step, we obtain the partial derivatives of the function Λ with respect to all network parameters.

Backpropagation Step:

The backpropagation step of the HMM network includes the partial derivative of the likelihood function (3.92) with respect to transition probabilities a_{ij} and emissions $p_i(D_t)$, which are calculated as

$$\frac{\partial \Lambda}{\partial a_{ij}} = \sum_{t=1}^{T} \frac{\partial \Lambda}{\partial c_t} \frac{\partial c_t}{\partial \ddot{\alpha}_t(i)} \frac{\partial \ddot{\alpha}_t(i)}{\partial a_{ij}}, \tag{3.93}$$

$$\frac{\partial \Lambda}{\partial a_{ij}} = \sum_{t=1}^{T} c_t \hat{\alpha}_{t-1}(i) p_j(D_t), \tag{3.94}$$

$$\frac{\partial \Lambda}{\partial p_i(D_t)} = \sum_{t=1}^{T} \frac{\partial \Lambda}{\partial c_t} \sum_{j=1}^{S} \frac{\partial c_t}{\partial \ddot{\alpha}_t(j)} \frac{\partial \ddot{\alpha}_t(j)}{\partial p_i(D_t)}, \tag{3.95}$$

$$\frac{\partial \Lambda}{\partial p_i(D_t)} = \sum_{t=1}^{T} \sum_{j=1}^{S} c_t \hat{\alpha}_{t-1} a_{ji}. \tag{3.96}$$

Derivatives obtained from Equation (3.96) are used for updating two parameter sets of the base demand vector $\mathbf{B} = [B_1, B_2, \ldots, B_S]$ and the weights of the emission network $\mathbf{W}^E$

$$\mathbf{B}_{n+1} = \mathbf{B}_n + \gamma_1 \left(\frac{\partial \Lambda}{\partial p_i(D_t)} \frac{\partial p_i(D_t)}{\partial \mathbf{B}_n} \right). \tag{3.97}$$

The input associated with base demand is the unit vector as in Figure 3.17. Therefore, the second partial derivative of $\frac{\partial p_i(D_t)}{\partial B_n} = 1$.

Regarding the update of the emission network weights we have

$$\mathbf{W}^E_{n+1} = \mathbf{W}^E_n + \gamma_1 \left(\frac{\partial \Lambda}{\partial p_i(D_t)} \frac{\partial p_i(D_t)}{\partial \mathbf{W}^E_n} \right), \tag{3.98}$$

where, the first partial derivative is obtained in (3.96) and the second one, partial derivatives of emissions $p_i(D_t)$ with respect to the network weights, is related to the backpropagation step in emission network $\mathbf{W}^E_n$, which is explained in Appendix A.

The newsvendor network is a feed-forward network that is trained by a back-propagation algorithm and the weights are updated using the derivatives of the cost

function with respect to $\mathbf{W}^{NV}$

$$\mathbf{W}_{n+1}^{NV} = \mathbf{W}_{n}^{NV} - \gamma_2 \left(\frac{\partial NVC}{\partial \mathbf{W}_{n}^{NV}} \right). \tag{3.99}$$

Further details of the partial derivative term are provided in Appendix A.

The transition matrix A is the joint part of the HMM and the newsvendor networks. Consequently, both partial derivatives of which in Equation (3.94) plus the partial derivative of the newsvendor cost function with respect to A are used to update the transition matrix

$$A_{n+1} = A_n - \gamma_3 \left(-\eta \frac{\partial \Lambda}{\partial A_n} + (1-\eta) \frac{\partial NVC}{\partial A_n} \right), \tag{3.100}$$

where the term in parenthesis is the partial derivative of the single objective function (P-HMMNV) with respect to A

$$A_{n+1} = A_n - \gamma_3 \left(\frac{\partial \text{Loss}}{\partial A_n} \right). \tag{3.101}$$

Network Specification:

We consider two hidden layers in the newsvendor network following the rule proposed by Huang (2003) that determine the number of hidden nodes in each layer. Further, we consider a single hidden layer for the emission network in which the number of hidden nodes is specified based on the formula suggested by Ke and Liu (2008). In training procedure of the HMMNV model, we choose the candidates for the trade-off coefficient η from the set of $\{0.001, 0.01, 0.1, 0.9, 0.99, 0.999\}$. To find the best learning rates for each network, a grid search over the set $\{0.001, 0.01, 0.1, 2, 10\}$ is used for all learning rates. We choose the best candidate among these parameters based on a cross-validation step on the training data set. We then train and test the model on all data set using the best parameter chosen by cross-validation.

Figure 3.18 indicates how the sample observations are divided into different sets so that one can implement the algorithms. First, we pick a smaller set of the data

and divide it into training and test samples for inner cross-validation to find the best parameters. We then train the algorithm with the best-selected parameter on the entire smaller set that we chose initially and test the model on the other test set to evaluate the performance of the model.

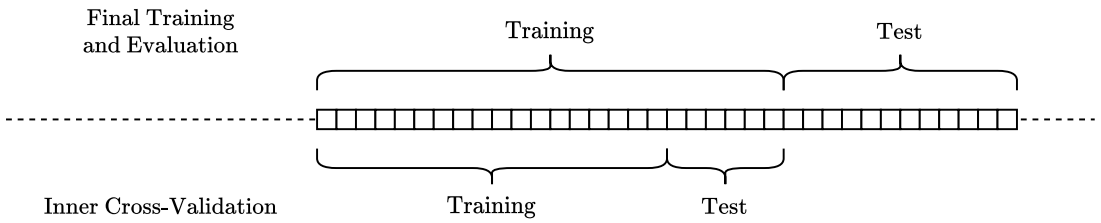


Figure 3.18: Different sets of the sample observations used for cross-validation, training, and test.

In the training procedure of the suggested model, we use Algorithm (1), which uses several hyperparameters including learning rates and the trade-off coefficient. Amongst the set of candidates for the trade-off coefficient, we choose the best candidate based on a cross-validation step on the training data set. Algorithm (2) shows the final train and test of the model on all data set using the best parameter chosen by cross-validation.

We then train algorithm 1 with the best-selected parameter on the entire smaller set that we chose initially and test the model on the other test set to evaluate the performance of the model.

3.7 Analyses

In this section, we test the performance of the HMMNV model using simulated data. In Section 3.2.3, we introduced several alternative methods as benchmarks. In Section 3.7.1, we describe our numerical setup to evaluate the performance of the HMMNV model with different benchmarks, and present and highlight the results in Section 3.7.2.

Algorithm 1 HMMNV Implementation

Require: $TRAIN \leftarrow$ Train Data, $TEST \leftarrow$ Test Data

Initialization: $\gamma_1 \leftarrow g_1$, $\gamma_2 \leftarrow g_2$, $\gamma_3 \leftarrow g_3$, $\eta_1 \leftarrow eta$, $BestNVCost \leftarrow$ inf, $A \leftarrow A_0$, $\pi \leftarrow \pi_0$, $\pi \leftarrow \pi_0$, $W^E \leftarrow W_0^E$, $W^{NV} \leftarrow W_0^{NV}$

1: $i \leftarrow 1$
2: **while** $|diff1| > \tau_1 \parallel |diff2| > \tau_2$ **do**
3: $\hat{E} \leftarrow \psi^E(F, W^E)$
4: $\hat{S} \leftarrow \alpha(\hat{E})$
5: $\hat{\Lambda} \leftarrow \Lambda(\hat{S})$
6: $diff1 \leftarrow \Lambda_{i+1} - \Lambda_i$
7: $W^E \leftarrow W^E + \gamma_1 \frac{\partial \Lambda}{\partial W^E}$
8: $\hat{F} \leftarrow [F, \hat{S}]_{TRAIN}$
9: $Q_{TRAIN} \leftarrow \psi^{NV}(\hat{F}, W^{NV})$
10: $NVC \leftarrow C(D_{TRAIN}, Q_{TRAIN})$
11: $diff2 \leftarrow NVC_{i+1} - NVC_i$
12: $W^{NV} \leftarrow W^{NV} + \gamma_2 \frac{\partial NVC}{\partial W^{NV}}$
13: $A \leftarrow A + \gamma 3 \cdot \left(-\eta \frac{\partial \Lambda}{\partial A} + (1-\eta)\frac{\partial NVC}{\partial A}\right)$
14: $i \leftarrow i + 1$
15: **end while**
16: $Q_{TEST} \leftarrow \psi^{NV}(\hat{F}_{TEST}, W^{NV})$
17: $NVC \leftarrow C(D_{TEST}, Q_{TEST})$
18: **if** $NVC < BestNVCost$ **then**
19: $\gamma_1^* \leftarrow \gamma_1$
20: $\gamma_2^* \leftarrow \gamma_2$
21: $\gamma_3^* \leftarrow \gamma_3$
22: $\eta^* \leftarrow \eta$
23: **end if**

Algorithm 2 HMMNV Inner Cross-Validation and Test

1: **for** $\eta \in \{$Good candidates$\}$ **do**
2: Train Algorithm 1 on smaller part of the training data set
3: Find the Newsvendor cost of the smaller out of sample set
4: **end for**
5: Choose the best η based on the cost obtained in line 3
6: Train Algorithm 1 on all data based on η^*
7: Find out of sample Newsvendor cost

3.7.1 Numerical Experiments

In this section, we first present the experimental setup that we have used to evaluate the performance of our method compared to the benchmarks. In our experimental setup, the demand in each period is a normally distributed random variable. The mean of this distribution in each period is determined based on the state of a Markov chain and the number of randomly generated features for that period. Table 3.7 shows the domain for each parameter of the system. According to Figure 3.16, we evolve the base demand by a two-state Markov chain model, and the additional part of the demand is generated by using a neural network and feature data. The final demand is the summation of state-dependent and feature-dependant demand values. Parameter e relates to the transition probability between two states of the Markov chain. The transition matrix is then determined as

$$\begin{array}{c} \\ s_1 \\ s_2 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{pmatrix} 1-e & e \\ e & 1-e \end{pmatrix} \end{array}$$

where s_1 and s_2 indicate two states of the Markov chain. Regarding the base demand in each state, we assume that they have normal distribution with parameters (μ_1, σ_1) and (μ_2, σ_2), respectively. We change the parameters of the first state within the range specified in Table 3.7 and fix those of the second state on $(\mu_2 = 1, \sigma_2 = 0.5)$. N is the number of features observed before the realization of the demand. All the features are randomly drawn from the standard normal distribution. We consider three different network weights denoted by W. Each network weight is a random variable with the standard normal distribution. The structure of the network, including the number of hidden layers, makes the relation between the features and demand more complex as the number of hidden layers increases. In our experiments, we consider up to three layers denoted by L. We consider 10 hidden nodes in each hidden layer where a sigmoid function serves as the activation function. We also consider three different values for the ratio between the cost rates in the problem (b/h). Finally, the parameter

Table 3.7: The set of parameters used in the experimental setup.

Parameter	Description	Domain
e	transition probability	0.01 0.1 0.2
μ_1	mean of the base demand in state 1	1 2 3
μ_2	mean of the base demand in state 2	1
σ_1	standard deviation of the base demand in state 1	0.5 1
σ_2	standard deviation of the base demand in state 2	0.5
σ_ϵ	standard deviation of the demand by features	0.5
N	number of features	1 3 5
W	network weights	1 2 3
L	number of hidden layers	1 2 3
b/h	newsvendor cost parameters ratio	2 5 10
T	number of observations	200 400 800

T is the number of observations. We evaluate the methods on all 5832 combinations that these parameter sets provide.

3.7.2 Results

In the following, we give a summary of the results of our experiments. In pairwise comparisons, HMMNV outperforms DNN in 64 % of the cases. HMMNV outperforms EDD in 60 % of the cases. HMMNV outperforms all the methods in 37 % of the cases. EDD outperforms all methods in 15 % of the cases.

Figures 3.19 presents the results of these experiments. In this figure, each point on the plot corresponds to data with a certain parameter set where the x-axis is labeled by the algorithms and the value of the y-axis represents the percentage deviation from the cost of the true model associated with each parameter set. A box plot is also depicted on the scattered result points of each algorithm. HMMNV obtained a lower mean and deviation from the mean compared to other methods. ObBW and EDD do not seem to perform well. The other three methods PfLR, LML, and DNN perform similarly. This grouping in performance implies that both sources that affect demand including observable features and non-observable Markov states have to be taken into account.

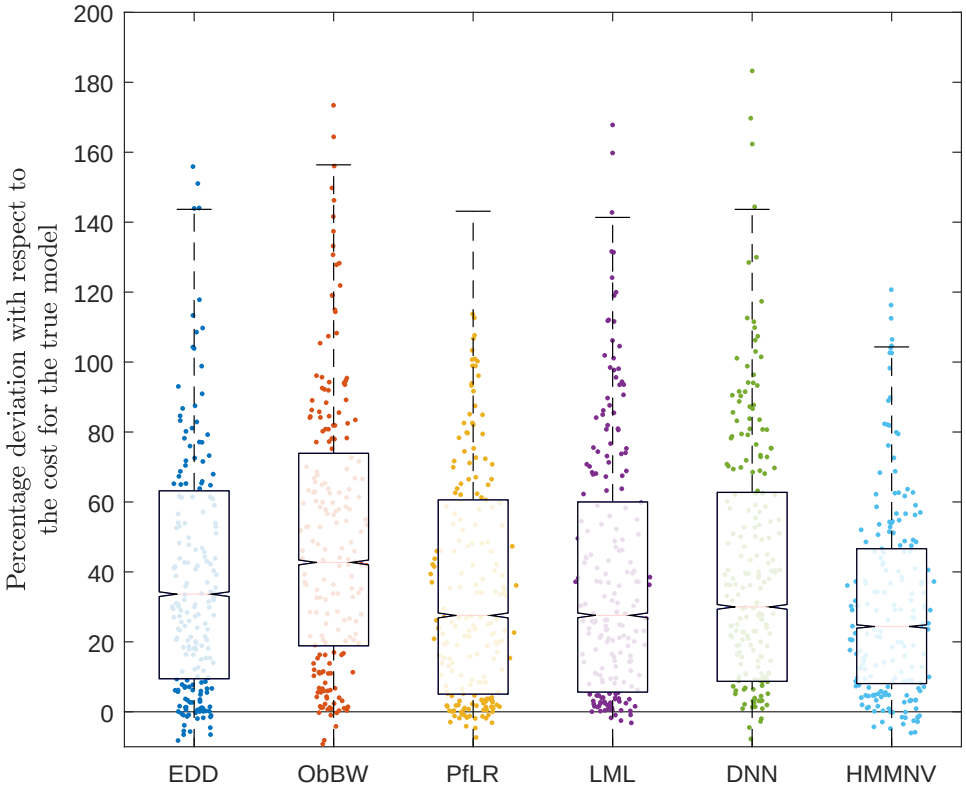


Figure 3.19: The result of all algorithms for different parameter sets.

To show the outperformance of each method over the others, we also implement a t-test on cost values two by two. This test proves if there is a meaningful and statistically significant difference between the results of any two algorithms. Table 3.8 shows the p-values of the test. If the algorithm in row i has a lower cost than the algorithm in column j, the cell ij of the table is filled by their p-value obtained from the test otherwise it is left empty. A p-value lower than 1 % indicates that the mean cost of an algorithm is less than the other one at a 1 % significance level. The HMMNV has a lower cost compared to others and the p-values close to zero confirm this. EDD as the simplest method outperforms only the ObBW. ObBW has shown higher costs compared to all other methods. PfLR and LML are very close and they outperform the DNN. DNN is better than EDD and ObBW at 5 % and 1 % levels,

Table 3.8: p-values of the t-test between the cost of algorithms.

	EDD	ObBW	PfLR	LML	DNN
EDD		3.7894e-100			
ObBW					
PfLR	7.7860e-13	9.8089e-168		0.8860	1.0743e-06
LML	3.5083e-12	1.1297e-163			2.6823e-06
DNN	0.0312	2.7515e-115			
HMMNV	1.6786e-80	4.6684e-299	2.4323e-35	1.5093e-35	9.6652e-63

respectively.

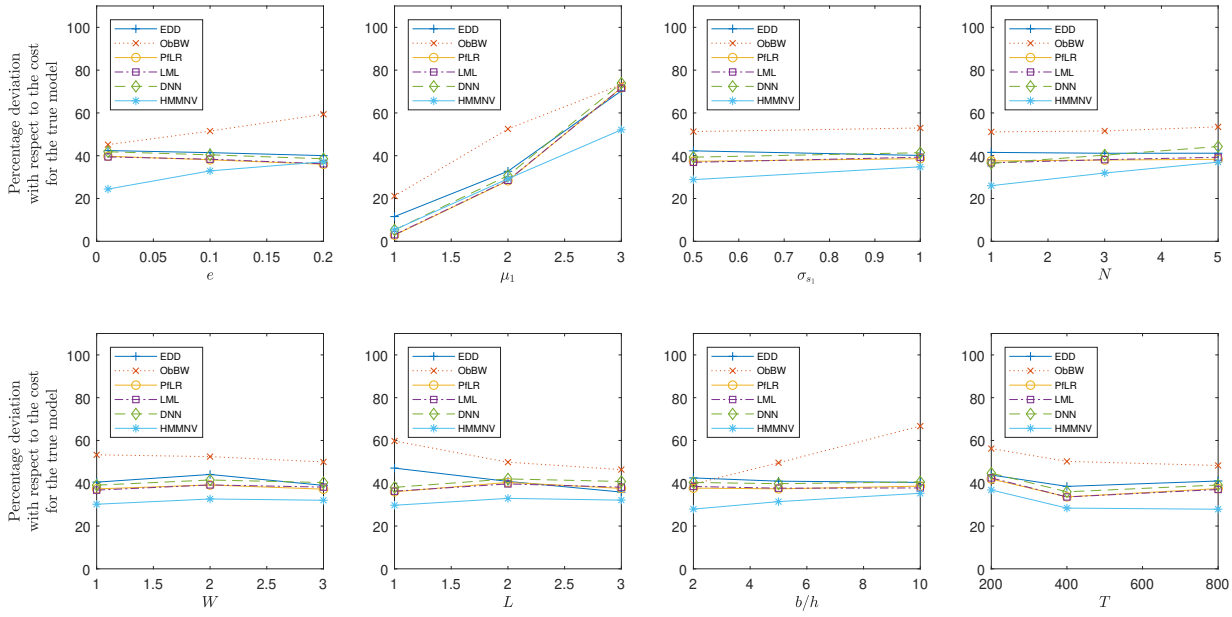


Figure 3.20: Deviations of algorithms' cost from the true model against the range of each parameter.

To investigate the effect of each parameter on the results and show how the algorithms differ over the range of the parameters, we plot the deviations of the methods for each parameter separately. Figure 3.20 represents the results for parameter values in table 3.7. The parameter e which refers to the transition probability of the hid-

den Markov is a representative for the information about the non-observable states. A lower value of e indicates more stability for the Markov chain and consequently greater importance and the effect caused by states on demand. As a result, we can see that the HMMNV method has a lower cost for the smallest value of e. As e increases the methods except ObBW converges which means that the systematic effect of Markov states on the randomness of the demand decreases. The parameter associated with the mean of the base demand in two states affects the results more than other parameters. As μ_1 increases, the long-term dependencies influence the demand more than observable features. As a result, HMMNV performs better than the benchmarks in cases with a higher imbalance in base demands. Another parameter of the base demand which is the standard deviation is not effective as of as the mean and all methods except HMMNV are insensitive to that. Similar to the mean of the base demand, standard deviation imposes more costs on our method as its value in one state differs more from the other state. The fourth plot shows the effect of the number of observable features indicated by N. It is reasonable that methods which don't use features like EDD and ObBW are not sensitive to N. Other methods converge as N increases. This results from the noises that features add to the demand. If features compose the underlying pattern, one can get better results by incorporating them into the model. A specific value of the parameter W which indicates the relations between features and demand is not meaningful compared to its other values but lower deviations over all W show the power of the HMMNV algorithm in achieving optimal policy with different random relations compared to others. The complexity of these relations stems from the structure of the networks and the associated number of hidden layers. It is observed that HMMNV results in lower costs regardless of the nonlinear complexity. The outperformance of the HMMNV method is obvious over different cost imbalances imposed by the b/h ratio. Deviations increase as this ratio increases in all methods. Regarding the number of observations, the results represent that our method works better as T increases. This is derived by the Markov sequence which is well captured using more observations. Briefly, all these analyses prove the

robustness of the suggested algorithm that results in a lower average cost with respect to other methods.

3.8 Real Data Experiment

We evaluate the performance of our algorithm and the benchmark methods using real data of the U.S. weekly crude oil demand. One can think of a product whose demand is closely correlated with U.S. crude oil. Data consists of weekly demand and covers the period from January 1986 to September 2020[1]. The series of the demand is shown in Figure 3.21.

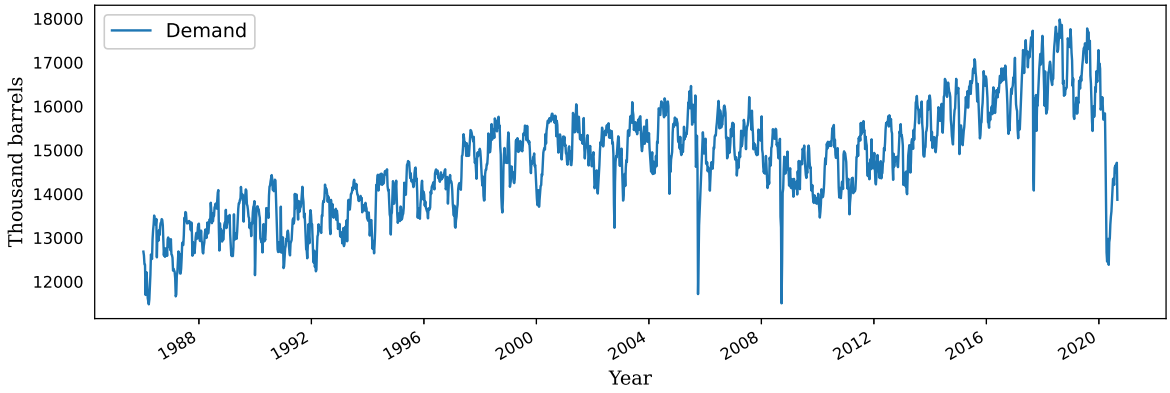


Figure 3.21: Time series of the weekly demand for crude oil from January 1986 to August 2020. The average demand is 14731 thousand barrels per week.

The set has 1800 weekly data sample observations. The set of features for the model includes 16 dummy variables representing the quarter of the year and the month of the year. These features capture the observable variation in demand which derives the seasonality. However, the variations caused by environmental randomness are not observable, we assume that they exist and follow a Markov chain model with two states. Our algorithm uses the feature data to explain the additional demand and model the residuals as base demand which are not captured by features. Base

[1]Data is collected from the U.S. Energy Information Administration at: `https://www.eia.gov`

demand is modeled by the Markov chain and evolves the demand over time along with the additional demand.

We pick a time window with the length of 500 weeks and train the algorithms on the first 400 weeks and test the trained models on the remaining 100 weeks. We roll this window by 100-week forward steps and obtain 14 test periods in total. Table 3.9 summarizes the result of the algorithms. This table shows the average cost of each method over 14 test periods (1400 weeks) for each newsvendor cost parameters ratio (b/h).

Table 3.9: Average cost of the crude oil ordering policy by each algorithm (values are divided by 10^3).

b/h	EDD	ObBW	PfLR	LML	DNN	HMMNV
2	1.161	0.840	1.150	1.123	1.117	0.477
5	1.660	1.262	1.751	1.701	1.703	0.650
10	2.158	1.625	2.282	2.274	2.244	1.557
20	2.871	2.134	2.847	3.104	3.178	1.832

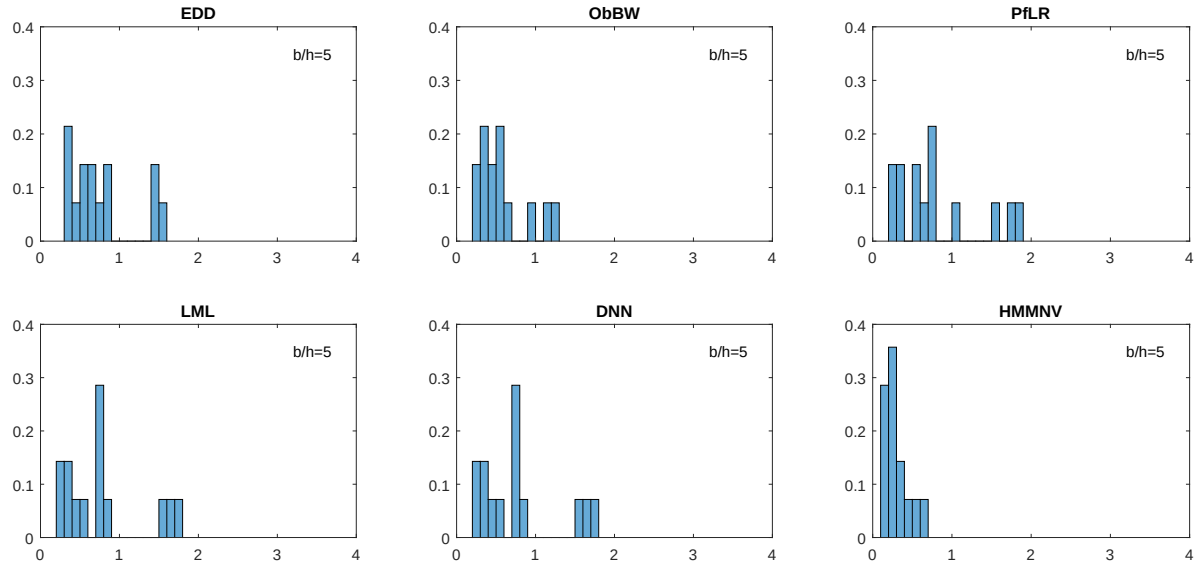


Figure 3.22: The frequency of the cost values obtained by each algorithm for the real data over all test periods.

It is observed that our proposed method outperforms other approaches for all values of cost parameters ratio. The second-best approach is ObBW that uses demand and fits a Markov model. Its outperformance suggests that the states of the environment have a long dependency and influence the demand in a state-dependent manner. The benchmark methods which use features have similar results with minor differences. Briefly, ObBW and EDD are still the best benchmarks among others. We conclude that if the randomness of the features and the environment are combined in a model, the cost decreases drastically.

Figure 3.22 shows the frequency of costs over 14 test periods for all methods. The suggested method offers an ordering policy that results in lower costs than other methods. To show the states obtained by the HMMNV method and the optimal policy, we plot the demand and the order quantity for 200 recent out of sample weeks in Figure 3.23. It is observed that the order quantity tracks the demand closely, imitating the present variations in the demand. The bottom panel of this figure shows the estimated sequence of the states during this period. When comparing the order and demand series to the states, we observe that the HMMNV algorithm assigns demands with higher variations to state 1 while the data labeled by state 2 has lower variations. Consequently, the order quantity by HMMNV has variations similar to that presented in the demand.

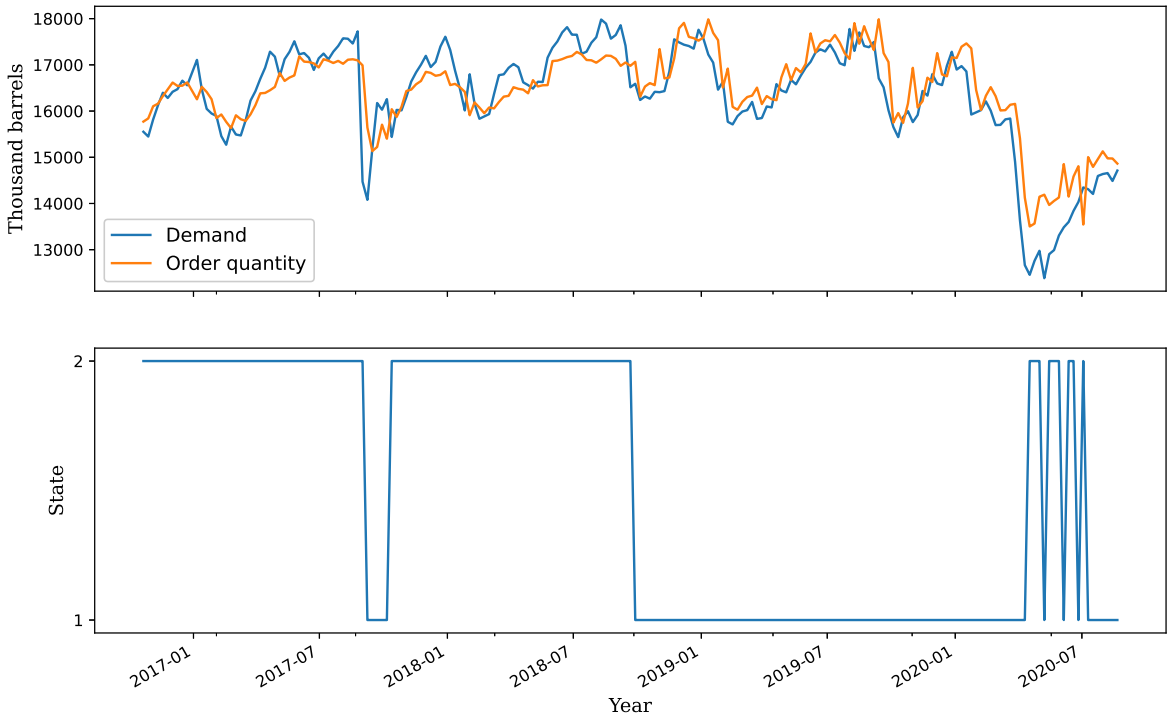


Figure 3.23: Optimal order quantity obtained by HMMNV for $b/h = 5$ during 200 weeks of test periods from October 2016 to August 2020 (top panel). Bottom panel shows the corresponding state sequence of the Markov model estimated by this algorithm.

3.9 Conclusion

The main contributions and findings of this chapter are as follows:

- An integrated solution method using neural networks is presented for a data-driven inventory system, which considers both observable and unobservable sources of features that affect the randomness in demand.

- Hidden Markov model as the most used modeling of the evolving environment is incorporated into a machine learning tool.

- The numerical results reveal that the proposed approach performs better than other methods when there might be unobservable features.

- The results present evidence that integrating unobservable feature estimation and inventory optimization is feasible and may bring significant improvements.

Chapter 4

STATE-DEPENDENT ASSET ALLOCATION

In this chapter, we focus on the asset allocation in the portfolio choice problem for a myopic investor who adjusts the portfolio weights periodically to hedge changes in the investment opportunity set. We first propose a state-dependent method which is a conditional strategy using neural networks to maximize the performance of the portfolio in terms of different performance ratios. Then, in section 4.10, we extend the model for the investors with state-dependent preferences and the objective of final wealth.

4.1 Problem Description

The mean-variance framework of Markowitz (1952) has been widely recognized as the foundation of modern portfolio theory. Inspired by this framework, investors traditionally consider a static single-period optimization for their asset allocation problem and optimize efficiency across all economic conditions. The allocation can be revisited regularly, but it is unlikely to change significantly, as long as the aim is efficiency across all market conditions (Nystrup et al., 2018). However, economic and market conditions change over time and present time-varying investment opportunity sets. Consequently, the means, variances, covariances, and higher-order moments of assets are time-varying. If economic conditions are linked to asset returns, a state-dependent asset allocation process should add value over static weights (Harvey, 2001). In this chapter, we first propose an approach using neural networks to directly decide on portfolio weights in the optimization problem. This approach estimates the optimal portfolio weights for a myopic investor at the beginning of the investment period without estimating a specific distribution of the data. In this approach, the investor's

ultimate objective is to maximize a performance ratio, such as the Sharpe ratio, as a mean-variance utility function.

However, maximization of performance measures has been widely used in the literature to find the optimal portfolio (Ortobelli et al., 2018), different optimal portfolio weights would be obtained due to variations in return moments if we optimize several ratios to take advantage of investment opportunities. When the optimal weights are used to translate the measure values into the realized performance of each optimal portfolio, we end up with different returns produced by the portfolios. To choose only one of these candidates, as the optimal weight, we pick the portfolio with the highest final wealth in a period. Accordingly, we assess the outcome of the portfolios obtained by different performance ratios and extend the network model to select the best one in a new period so that the final wealth would be the highest compared to other choices.

4.2 Literature Review

In the following, we provide a review of the pertinent literature. This is presented in two parts. First, we review the literature on state-dependent asset allocation with different performance ratios. Then, we review the related work on the time-varying investment preferences.

4.2.1 State-dependent Asset Allocation

In Finance, data have empirical distributions with heavy tails, time-varying moments, and multimodality. Parametric modeling available by classical methods ignores these important features of data or focuses on extreme events which lead to missing information about other parts of distributions (Rodriguez et al., 2008). In contrast to distributional modeling of returns which has several drawbacks, recent literature captures the dynamic behavior of returns over time by exploring the dynamics through the market state. The market state is represented by some time-varying and market-wide variables. Several studies have used these state variables to predict returns

better while raising the time-varying issue in moments (Merton, 1980; Fama and French, 1989; Diebold et al., 1994; Barberis, 2000; Aït-Sahalia and Brandt, 2001; Ang and Bekaert, 2004; Kanas, 2008; Chen, 2009; Ernst et al., 2009; Nystrup et al., 2018; Costa and Kwon, 2019). When we maximize a ratio that is made up of time-variant returns, the dynamics of returns will be transmitted into portfolio decisions. Hence, the decisions are affected by market state in the same way and therefore all the relationships between state variables and returns exist between state variables and the decisions.

The purpose of state-dependent asset allocation strategies is to take advantage of good economic and market conditions (or states) and mitigate the impact of adverse states (Sheikh and Sun, 2012). Several studies show the importance of state-dependent asset allocation to exploit the predictability of the moments of asset returns and hedge changes in the investment opportunity set in both single-period and multi-period contexts (Barberis, 2000; Aït-Sahalia and Brandt, 2001; Ang and Bekaert, 2004; Brandt and Santa-Clara, 2006; Guidolin and Timmermann, 2007; Kritzman et al., 2012; Laborda and Olmo, 2017; Nystrup et al., 2018). However, it is very difficult to model a conditional return distribution and find a closed-form solution for optimal investment strategies unless strong assumptions on the statistical distribution of asset returns are imposed. The predominant approach includes a two-step process (Laborda and Olmo, 2017). Firstly, modeling separate features of the return distribution, and then determining the optimal portfolio choice.

In classical approaches of the problem, the returns distribution (e.g., normal) is known and the parameters of the distribution are either known or estimated by some available data. In the portfolio problem, this approach follows the common notion about the returns distribution and supposes the returns have a normal distribution. The traditional method is then a two-step optimization (Chen et al., 2011). In the first step, the parameters of the returns are estimated. In the second step of the method, the estimated parameters are plugged into the model and the problem is optimized. Accordingly, the mean vector and covariance matrix are the parameters that have to

be estimated based on the normality of returns. Different approaches are suggested to obtain more precise predictions of return moments. Historical data and sample estimation are usually used to forecast the mean vector and covariance matrix. Several models such as Vector Autoregressive (VAR) model are evaluated to use for this goal in the literature (Kroll et al., 1984; Bekaert and Hodrick, 1992; Campbell et al., 2001; Chen et al., 2011). In addition to the sample estimations, factor-based models show outperformance over the estimation based on sample period (Chan et al., 1999). However, these approaches do not guarantee the optimal allocation for the problem at hand because different sets of state variables may predict different moments of returns used in the objective function (Aït-Sahalia and Brandt, 2001). Alternative approaches include numerical methods, and stochastic programming (Brennan et al., 1997, 1998; Campbell and Viceira, 1999; Barberis, 2000; Campbell et al., 2001, 2002, 2013) or semi non-parametric methods (Aït-Sahalia and Brandt, 2001; Brandt and Santa-Clara, 2006), which have limitations in the number of and type of state variables used in the process or require fitting a linear model with data. Nevertheless, the dependence of the optimal portfolio choice on the whole return distribution is very complex and far from linear and the predicting power of specific state variables may change over time.

A few research relates portfolio weights directly to state variables. For example, Aït-Sahalia and Brandt (2001) suggest a semi-non parametric method to form a linear index of state variables and select the best variables to directly predict the optimal portfolio weights. More recently, and Laborda and Olmo (2017) provide a linear parametric portfolio policy rule that focuses directly on the dependence of the portfolio weights on the predictor variables. As Aït-Sahalia and Brandt (2001) state that dependence of the optimal portfolio choice on the whole return distribution is so complex that only numerical methods can be used. Thus, along with taking advantage of market state variety, the complexity of modeling these relationships leads to the emergence of nonparametric methods with more reliability and accuracy (Butler and Schachter, 1997; Yao et al., 2013; Ortobelli et al., 2017).

The use of new statistical approaches and machine learning techniques in pattern recognition to construct optimal portfolios amends this challenge and facilitates efficient modeling of the relations between state variables and portfolio weights. In addition to modeling non-linear relations between input and optimal targets and decisions, neural networks can take many input data (Rojas, 2013). Therefore, the proposed DNN-based method accommodates an arbitrarily large number of state variables in the information set. In few related studies, external features rather than state variables are used in asset allocation and portfolio optimization in neural network settings. For example, Malandri et al. (2018) use public financial sentiment and historical prices to directly prescribe wealth allocation. They show that with several input variable data, neural networks efficiently learn the impact of the public mood on financial time series. Xing et al. (2018) use social media and text mining to translate sentiment information into market view and show that new forecasting techniques improve the portfolio performance. Our study complements this literature by proposing a DNN-based approach for state-dependent asset allocation in portfolio management.

4.2.2 Time-varying Preferences

Each performance ratio computes return and risks differently since they incorporate various moments of returns in calculations. Accordingly, using these ratios leads to different portfolios. Farinelli et al. (2008) provide evidence that emphasizes the issue of time-varying portfolio performances using different ratios and suggests that investors do not rely on a single measure for portfolio optimization. However, they do not suggest any consolidated approach to guide for the optimal asset allocation which takes into account the time variation of performance and market states. To the best of our knowledge, we are the first to address this issue and provide a method that identifies and uses different performance ratios, rather than one single ratio over time, depending on the state of the market. For instance, Biglova et al. (2004) use various performance measures including theirs in a single period, and show that asset

allocation based on their ratios leads to higher return compared to other measures. The well-known Sharpe ratio is valid for the portfolio choice of an investor if returns are normally distributed because mean and variance can explain their distribution; however, it is well documented that returns do not follow normal distributions (Leland, 1999; Kat and Brooks, 2001; Agarwal and Naik, 2004; Malkiel and Saha, 2005). When returns exhibit heavy tails or present kurtosis or skewness, using Sharpe as the objective function leads to an inefficient estimation of risk, and incorrect investment decisions. To overcome this well-known limitation of the Sharpe ratio, several alternative performance ratios have been proposed in the literature. For example, in Gini (Shalit and Yitzhaki, 1984), Mean Absolute Deviation (MAD) (Konno and Yamazaki, 1991) and Minimax (Young, 1998), the risk measures are redefined. Recently, asymmetrical parameter-dependent ratios have become popular. Value at Risk (VaR) and Conditional VaR (CVaR) (Martin et al., 2003) capture downside risk, and Rachev (Biglova et al., 2004) includes the truncated moments conditioned to tail events as the risk measure. However, the best fitting ratio changes over time. It is shown that time variations when using different measures also exist in evaluating assets. For example, Zakamulin (2010) and Caporin and Lisi (2011) show that different performance measures lead to not only different but also time-varying, rankings depending on the sample period. The performance measures do not give a consistent ranking outcome, since they use different moments, partial moments, or different selections of risk variables (Billio et al., 2015). This is because different risk measures in performance ratios to capture returns distribution and moments are used. This time-varying issue results in wrong optimal decisions when the portfolio manager uses a single measure for portfolio choice for all investment periods.

According to modern portfolio theories, performance ratios, as mean-variance utility functions, are assumed to be law-invariant (not state-dependent). This suggests that investors are only interested in the distribution of returns and their preferences are not adjusted based on the state of the economy in which returns are realized, which is at odds with the needs of many investors (Bernard et al., 2015b). Several

studies consider time-variant risk aversion, where preferences and utilities are affected by exogenous state variables and benchmarks (Merton et al., 1973; Cox et al., 1985; Gordon and St-Amour, 2004). Gordon and St-Amour (2000) show that models with regime-switching in preferences and beliefs rationalize the risk aversion puzzle. Most investors invest and then calibrate the preferences and infer the level of risk aversion according to market conditions. This consideration leads to a portfolio policy that is not solely determined by return target but is also based on interaction with the market (Bernard et al., 2015a, 2018). Performance ratios cover a range of preferences and show different levels of risk aversion.

4.3 Performance Ratios

In this section, we describe different common performance ratios for asset allocation and explain how reward and risk are considered in each one.

Sharpe

Sharpe (1966) ratio takes the expected returns of assets in a portfolio as the reward, and considers the standard deviation of the portfolio return as the risk. Hence, $\phi(r_p) = E[r_p]$ and $\rho(r_p) = E[(r_p - E[r_p])^2]^{\frac{1}{2}}$, then the Sharpe ratio is:

$$\psi_S(r_p) = \frac{E[r_p]}{\sigma_{r_p}} \tag{4.1}$$

Sharpe ratio uses the first and second moment of returns. Since a normal distribution is specified by the two first moments, this ratio is suitable if returns follow normal distribution.

Other ratios except Rachev ratio presented hereafter, are mainly different in terms of the risk definition with the same reward function as in Sharpe.

MAD

Konno and Yamazaki (1991) define the risk by the absolute distance of portfolio return from its mean. In this ratio, the outliers have less effect than in standard deviation where the distances are squared. So, the MAD ratio measures the portfolio

performance as follows:

$$\psi_{MAD}(r_p) = \frac{E[r_p]}{\rho_{MAD}(r_p)} \tag{4.2}$$

The risk measure in this ratio is:

$$\rho_{MAD}(r_p) = E[|r_p - E[r_p]|] \tag{4.3}$$

MiniMax

Young (1998) develops linear programming similar to the mean-variance principle of Markowitz for the portfolio choice. His model uses the minimum of returns of a portfolio during a period as a measure of risk. If the minimum return of a portfolio is $\underline{M} = \min_{t\in\{1,D\}} r_{p_t}$, the MiniMax approach solves the problem to obtain the optimal weights of $\mathbf{x}^*$ by maximizing $\underline{M}$. Hence, the following optimization gives the portfolio decision:

$$\mathbf{x}^* =: \arg\max_{\mathbf{x}\in\mathcal{X}} \ \underline{M} \tag{4.4}$$

If we consider a level of return that is acceptable by an investor and denote it by $\underline{r}$, maximizing the MiniMax risk measure means that the investor seeks a portfolio that minimizes the maximum loss, $l_{max} = \underline{r} - \underline{M}$, over all past observations in a period. To be consistent with the definition of the risk measure in MiniMax ratio in terms of minimizing the risk, we consider $\rho_{MiniMax}(r_p) = -\underline{M}$. Finally, the ratio is formulated as follows:

$$\psi_{MiniMax}(r_p) = \frac{E[r_p]}{\rho_{MiniMax}(r_p)} \tag{4.5}$$

Gini

Gini's mean difference is an index to measure the variability of a random variable. It is calculated based on the expected value of the absolute difference between every pair of the variable realization. Let $F(r)$ and $f(r)$ denote the cumulative distribution

and density function of variable r, respectively. There exist bounded values of a and b such that $F(a) = 0$ and $F(b) = 1$. The Gini's mean difference is defined as follows:

$$\Gamma = \frac{1}{2} \int_a^b \int_a^b |r_t - r_{t'}| \, f(r_t) f(r_{t'}) \, dr_t dr_{t'} \tag{4.6}$$

Here, r_t and $r_{t'}$ denote two realizations of variable r. Portfolio return is also a random variable. Given asset returns at times t and t', we can denote the portfolio return at these times by r_{p_t} and $r_{p_{t'}}$, respectively. Then, similarly we write the Gini formula for the portfolio as below:

$$\Gamma_p = \frac{1}{2} E \left[|r_{p_t} - r_{p_{t'}}| \right], \quad \forall t, t' \in T, \; t \neq t' \tag{4.7}$$

Having a new measure for dispersion of portfolio return which denotes the risk, the Gini performance ratio is expressed as below:

$$\psi_{Gini}(r_p) = \frac{E\left[r_p\right]}{\rho_{Gini}(r_p)} \tag{4.8}$$

Here, $\rho_{Gini}(r_p) = \Gamma_p$.

CVaR

Value-at-risk (VaR) as a well-known risk measure, has some mathematical drawbacks which make it an unattractive measure for financial risks. For example, it is unable to take into account extreme events (loss/gain) which could have a heavier effect on analyzing a risky position. A descendant of VaR is then introduced as a risk measure that is consistent with financial concepts. Conditional Valu-at-risk preserves the desired properties for a risk measure.

Definition 4.3.1. *Conditional Valu-at-risk (or Tail conditional expectation). Given an underlying probability measure $\mathbb{P}$ on Ω, a total return r on a reference instrument,*

and a level α, CVaR is the measure of risk defined by (Artzner et al., 1999)*:*

$$CVaR_\alpha(r) = -E\left[\, r \,|\, r \leq -VaR_\alpha(r)\right] \tag{4.9}$$

A new performance ratio using CVaR as the risk is formulated as follows:

$$\psi_{\mathrm{CVaR}}(r_p) = \frac{E\left[r_p\right]}{\mathrm{CVaR}_\alpha(r_p)} \tag{4.10}$$

Rachev

Fishburn (1977) defines Lower Partial Moments (LPM) for a random variable. This measure examines the moments of degree n below an acceptable threshold τ:

Definition 4.3.2. *If τ denotes an arbitrary origin as the reference level and n the degree of the moment which implies the risk aversion, and R is a random variable with cumulative distribution function $F_R(r)$, the LPM is given by:*

$$LPM_{n,\tau}(R) = \int_{-\infty}^{\tau} (\tau - r)^n \, dF_R(r) \tag{4.11}$$

Biglova et al. (2004) set the threshold in the Equation (4.11) to CVaR_α, called Expected Tailed Loss (ETL), and introduce a new risk measure. Their empirical analysis shows that a correct reward perception is also important to distinguish a desirable risky investment. The corresponding performance ratio then has a reward defined differently from previous ratios introduced so far. If r_p denotes the portfolio return, then $L = -r_p$ represents the relative loss. The new ratio is the ratio between the CVaR of the return at a given confidence level to the CVaR of the relative loss at another confidence level, Hence:

$$\psi_R(r_p) = \frac{ETL_\beta(r_p)}{ETL_\alpha(-r_p)} \tag{4.12}$$

Here, $ETL_\alpha(X) = E\left[X \,|\, X \geq \mathrm{VaR}_\alpha\right]$.

4.3.1 Properties of Performance Ratios

In the portfolio optimization problem, the properties of the whole function ψ are of interest rather than those of reward and risk separately. However, we shall investigate the functions ϕ and ρ to obtain properties of ψ. Several characteristics are considered for a risk measure from a financial standpoint. A risk measure is known as *coherent* if it satisfies the following axioms (Artzner et al., 1999):

Axiom (T). Translation invariance. *For all* $X \in \mathcal{G}$ *and all real numbers* ν, $\rho(X+\nu) = \rho(X) - \nu$.

Axiom (S). Subadditivity. *For all* X_1 *and* $X_2 \in \mathcal{G}$, $\rho(X_1 + X_2) \leq \rho(X_1) + \rho(X_2)$.

Axiom (PH). Positive homogeneity. *For all* $\beta \geq 0$ *and all* $X \in \mathcal{G}$, $\rho(\beta X) = \beta\rho(X)$.

Axiom (M). Monotonicity. *For all* X *and* $Y \in \mathcal{G}$ *with* $X \leq Y$, *we have* $\rho(Y) \leq \rho(X)$.

Since the risks in all the selected performance ratios are coherent, they are also convex based on the following Proposition 4.3.1 (Schmeidler, 1989):

Proposition 4.3.1. *If a risk measure has properties (S) and (PH), it is a convex risk measure.*

The rewards in all ratios except the Rachev ratio are the expectation of all asset returns which is an affine function. Affine functions are both convex and concave. Finally, by the convexity of risk measures and the concavity of reward function we conclude the quasi-concavity of performance ratios from the following Proposition:

Proposition 4.3.2. *If* ϕ *is concave and* ρ *is convex, then* ψ *satisfies the quasi-concavity property.*

Quasi-concavity of a performance ratio shows that an investor prefers diversification by choosing an average position rather than an extreme one. In the Rachev ratio, since the reward is a CVaR function, the numerator in this ratio is also a convex function. As a result, quasiconcavity is not guaranteed in the Rachev ratio. However, we would examine our method with this ratio to show the generality of the method in non-convex optimization.

4.4 State Variable

Some of variables on mean predictability include; term spread (Campbell, 1987; Fama and French, 1988, 1989; Ferson and Harvey, 1991), default spread (Fama and French, 1988, 1989; Keim and Stambaugh, 1986), Treasury bill yield (Fama and Schwert, 1977; Ferson and Harvey, 1991). The variables for predicting variances include; lagged squared return and/or lagged variance (Bollerslev, 1986; Engle, 1982; French et al., 1987; Harvey, 2001; Schwert, 1989; Whitelaw, 1994), default spread (Harvey, 2001; Whitelaw, 1994), dividend yield (Harvey, 2001), debt-to-equity ratio (Schwert, 1989). Finally, the predictability of convarinces is attributed to some variables such as; lagged covariances, lagged cross-products of returns (Bollerslev et al., 1988), term spread (Campbell, 1987; Harvey, 2001), and default spread, and dividend yield (Harvey, 2001).

4.5 Portfolio Optimization Problem

In this section, we focus on the portfolio optimization problem by maximizing reward-risk ratios which are characterized by two elements. One, which is desirable, refers to reward and a rational investor seeks to increase it in her portfolio. The other refers to the uncertainty of the portfolio return which is not desirable and causes risk, so it should be minimized by a suitable composition of assets in the portfolio. In particular, reward-risk performance measures integrate these elements in a ratio so that a greater ratio offers a higher performance per unit of risk.

4.5.1 Predicting Optimal Portfolio Weights

We consider a single-period investor who maximizes the performance $\psi(r_p)$ of a portfolio with return r_p. The expectation is conditional on a vector of state variables $\mathbf{z}$. The optimal portfolio weights can be obtained by solving the following optimization

problem:

$$\begin{aligned} \max_{\mathbf{x}} \quad & E_{\mathbf{r}}\left[\psi\left(\mathbf{x}'\mathbf{r}\right)|\mathbf{z}\right] \\ \text{s.t.}: \quad & \mathbf{x}' \cdot \mathbf{e} = 1 \\ & \mathbf{x} \geq 0 \end{aligned} \tag{P1}$$

Here, $\psi\left(r_p\right) = \frac{\phi\left(r_p\right)}{\rho\left(r_p\right)}$ is a performance ratio, $\phi\left(r_p\right)$ and $\rho\left(r_p\right)$ denote the reward and risk of the portfolio return $r_p = \mathbf{x}'\mathbf{r}$, respectively. $\mathbf{x}' = (x_1, \ldots, x_N)$ denotes the vector of portfolio weights whose elements correspond to each of N assets in the portfolio, and $\mathbf{r}$ is the matrix in which any column is the vector of returns of each asset. The first constraint describes that no leverage is permitted with $\mathbf{e}$ presenting an N-dimensional unit vector, and the second constraint of nonnegativity shows that short selling is not allowed. The expectation in the objective function is taken over returns. Since the returns are random variables, they make the objective function noisy and stochastic rather than deterministic over different periods of investment. Therefore, stochastic problems are solved in the expectation of the objective function.

4.6 Predicting Optimal Portfolio Weights Using DNN

We use the Lagrangian multiplier method to solve Problem (P1). Introducing new variables of μ and $\boldsymbol{\lambda}$, the Lagrangian function is written as follows:

$$\mathcal{L} = E_{\mathbf{r}}\left[\psi\left(\mathbf{x}'\mathbf{r}\right)|\mathbf{z}\right] + \mu(\mathbf{x}' \cdot \mathbf{e} - 1) - \boldsymbol{\lambda}'\mathbf{x} \tag{P2}$$

Here, the variables in the form of a vector or matrix are indicated by boldface letters. We introduce vectors of new variables of μ and $\boldsymbol{\lambda}$. Now, problem (P2) is converted to an unconstrained optimization problem. The following system of equations which shows the first-order necessary conditions are called Karush-Kuhn-Tucker (KKT) conditions and gives the candidate points for a maximum (Bazaraa

et al., 2013):

$$\begin{aligned} \nabla_{\mathbf{x}}\mathcal{L} &= 0 \\ \mathbf{x}' \cdot \mathbf{e} - 1 &= 0 \\ \boldsymbol{\lambda} \odot \mathbf{g}(\mathbf{x}) &= 0 \\ \mathbf{x} &\geq 0 \\ \boldsymbol{\lambda} &\geq 0 \end{aligned} \tag{4.13}$$

Here, ∇ is the vector of partial derivatives, and $\odot$ is an element-wise multiplication operator. We know that the function f is quasi-concave and all the constraints are linear in a portfolio optimization problem of the form (P2), so the following result is useful in this case:

Proposition 4.6.1. *If there exist a unique vector* $\boldsymbol{\lambda}$ *such that* $(\mathbf{x}^*, \boldsymbol{\lambda})$ *satisfies the system* (4.13)*, and it is not the case that* $\nabla_{\mathbf{x}}\psi(\mathbf{x}^*) \neq 0$ *then* $\mathbf{x}^*$ *solves Problem* (P2)*.*

Proof of Proposition 4.6.1. Using the assumption of quasi-concavity of f we conclude that:

$$\nabla_{\mathbf{x}} f(\mathbf{x})'(\mathbf{x}^* - \mathbf{x}) \geq 0 \tag{4.14}$$

Now suppose, in contrary to the optimality of the point $\mathbf{x}^*$, there exists a point $\mathbf{x}$ which satisfies the constraint and $f(\mathbf{x}) > f(\mathbf{x}^*)$. So, we can pick $\epsilon > 0$ and sufficiently small to find a new point $y = x - \epsilon\nabla_{\mathbf{x}} f(\mathbf{x}^*) - \mathbf{x}^*$ such that $f(y) > f(\mathbf{x}^*)$. Thus:

$$\begin{aligned} &\nabla_{\mathbf{x}} f(\mathbf{x}^*)(x - \epsilon\nabla_{\mathbf{x}} f(\mathbf{x}^*) - \mathbf{x}^*) \\ &\qquad = \nabla_{\mathbf{x}} f(\mathbf{x}^*)(x - \mathbf{x}^*) - \epsilon\|\nabla_{\mathbf{x}} f(\mathbf{x}^*)\|^2 < 0 \end{aligned} \tag{4.15}$$

Given the assumption that $\nabla_{\mathbf{x}} f(\mathbf{x}^*) \neq 0$, above inequality implies that $f(x - \epsilon\nabla_{\mathbf{x}} f(\mathbf{x}^*)) < f(\mathbf{x}^*)$, contradicting $f(\mathbf{x} - \epsilon\nabla_{\mathbf{x}} f(\mathbf{x}^*)) > f(\mathbf{x}^*)$. Thus, in fact, $f(\mathbf{x}) < f(\mathbf{x}^*)$.

In this study, we consider a vector of state variables indicating the market condition to be able to solve the stochastic portfolio problem. The vector of state variables

represents an element in the set of all states of nature, Ω, at the beginning of each period t. The probabilities of various states occurring may be unknown and we assume that the distribution of the returns depends on the state of the market. As it is known that selected state variables predict the return moments, we expect this relationship to exist between state variables and portfolio decisions as well. Moreover, we assume the time homogeneity of the conditional return distribution given the state variables. To incorporate state variables into the Lagrangian function, we consider function Λ which takes state variable vector $\mathbf{z}$ and returns the portfolio choice $\mathbf{x}$:

$$\mathbf{x}_t = \Lambda(\mathbf{z}_t) \tag{4.16}$$

We assume that the function Λ is not a time-variant; however, the time variations of the state variable vector impose the required optimal variations in portfolio choice.

Now we provide a fully nonparametric estimation for $\Lambda(\mathbf{z}_t)$ using DNN following Bradrania and Neghab (2021) who provide a complex function Λ that gives the portfolio choice using the state variables. In our proposed DNN, the inputs are state variables, and the sigmoid function is used as the activation function (in the hidden layer). The constraints of Problem (P1) impose $\mathbf{x} \in [0, 1]$, so we use a sigmoid function in the output nodes that aggregate the outputs from the hidden layer and generates values between 0 and 1 in the output nodes of the network (as the portfolio weights). This allows the non-negativity constraint ($\mathbf{x} \geq 0$) to be satisfied automatically, so the Lagrangian Function (P2) reduces to Equation (4.17) for one historical sample period t (e.g., a month):

$$\mathcal{L} = E_t \left[\psi \left(\mathbf{x'}_t \mathbf{r}_t \right) | \mathbf{z}_t \right] + \mu(\mathbf{x}'_t \cdot \mathbf{e} - 1) \tag{4.17}$$

Here, $\mathbf{z}_t \in \mathbb{R}^M$ and $\mathbf{r}_t \in \mathbb{R}^{D_t \times N}$, and D_t denotes the number of working days in period t for $t = 1 \ldots, T$. All observations for T periods can be represented as $\{(\mathbf{z}_1, \mathbf{r}_1), \ldots, (\mathbf{z}_T, \mathbf{r}_T)\}$. We substitute portfolio weights $\mathbf{x}_t$ with $\Lambda(\mathbf{z}_t)$ which is the output of the DNN with inputs of state variables $\mathbf{z}_t$.

Here, $\mathbf{x}$ is the vector of decision, $\mathbf{r}$ is a random vector of asset returns included in the portfolio, and $\mathbf{z}$ is a random vector of observed state, all associated with one period. Hence, we first observe a random state $Z \in \mathcal{Z}$ which affects the distribution of $\mathbf{r}$ and the function $\mathcal{L}$. When we obtain a position for the portfolio using a decision $\mathbf{x}$, we would observe a noisy function induced by an observed random variable $\mathbf{r}$. Similarly to probability $\mathbb{P}$, the conditional distribution of $p(\mathbf{r}|Z = \mathbf{z})$ is unknown. Therefore, we use samples of historical scenarios, $\omega_1, \ldots, \omega_T$, all in Ω, coming from different periods. Therefore, we optimize the expectation of the Lagrangian functions overall observations as an approximate of the Lagrangian of Problem (P1). The goal of the DNN is to maximize the following function:

$$\mathcal{L}_t = E_t \left[\psi \left(\Lambda(\mathbf{z}_t)' \mathbf{r}_t \right) + \mu(\Lambda(\mathbf{z}_t)' \cdot \mathbf{e} - 1) | \mathbf{z}_t \right] \tag{4.18}$$

which is an average of functions $\mathcal{L}_t$'s for all historical sample observations over the periods $t = 1, \ldots, T$.

This is the objective function in our DNN, which is optimized to find the optimal weights of the portfolio. To maximize the final objective Function (4.18) by the DNN, we need to incorporate the Lagrangian multiplier μ into the network as an additional network weight. In the optimization algorithm, we use the gradient descent approach, which follows the positive direction of the gradients to optimize Function (4.18). Fortunately, there exists a range of iterative algorithms which have great empirical success for this optimization problem. The most popular method to solve conditional stochastic optimization problems is the stochastic approximation using stochastic gradients (Hannah et al., 2010). The main idea in this method is to optimize a function iteratively by generating a sequence of solutions based on random gradients until there is no improvement in the objective function or some other criteria are met. The following equation shows the iteration:

$$\mathbf{x}_{n+1} = \Lambda_{\mathcal{X}}(\mathbf{x}_n + \gamma \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}_n, \mathbf{r}(\omega_{n+1})) \tag{4.19}$$

Using this algorithm, we optimize the Lagrangian function iteratively by generating a sequence of network weights based on random gradients until there is no improvement in the function.

Since $\partial L/\partial \mu = \mathbf{x}' \cdot \mathbf{e} - 1$ the algorithm does not end as long as $\mathbf{x}' \cdot \mathbf{e} \neq 1$. Thus, as the function value converges, we are sure that the budget constraint is satisfied, i.e. $\mathbf{x}' \cdot \mathbf{e} = 1$. Once the network is trained (after the last iteration), we have the predicting model ($\hat{\Lambda}(\mathbf{z}_t)$) that enables us to obtain portfolio weights for a new period, directly, from the network's inputs (i.e. state variables) without any return prediction separately. The DNN, as the predicting model to find the optimal weights given market states, is shown in Figure 4.1.

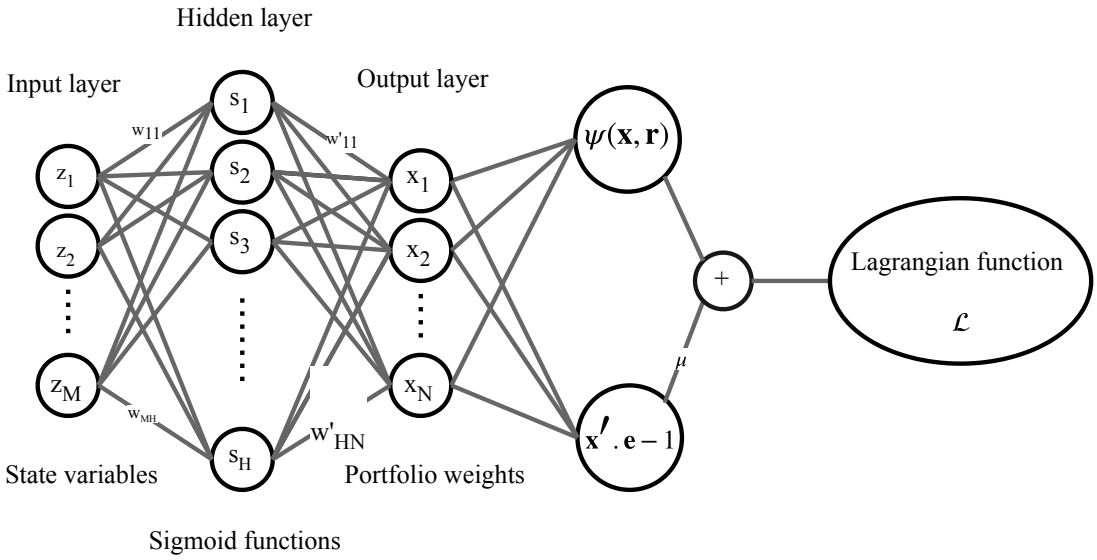


Figure 4.1: Structure of the Deep Neural Network for ratio maximization. The DNN network, as the predicting model, to find the optimal weights given market states.

Algorithm (3) provides the training procedure of the proposed DNN based on the iterations required in the optimization algorithm.

Algorithm 3 DNN training for the training period

Start
With the ratio ψ **do**
W, $\mu \leftarrow$ random initial weights for the Network Λ
while stopping criterion is not met **do**
  for $t \leftarrow 1$ **to** T **do**
    observing the vector of state variables $\mathbf{z}_t$
    $\mathbf{x}_t \leftarrow \Lambda(\mathbf{z}_t)$, obtaining portfolio weights from the network
    $r_{p,t+1} \leftarrow \mathbf{x}'_t \mathbf{r}_{t+1}$, computing portfolio returns using realized returns $\mathbf{r}_{t+1}$
    $\mathcal{L}_t$, computing the Lagrangian function
end for
calculating the objective function $\hat{\mathcal{L}} = \frac{1}{T}\mathcal{L}_t$ using all $\mathcal{L}$
finding partial derivatives $\frac{\partial \hat{\mathcal{L}}}{\partial W}$ and $\frac{\partial \hat{\mathcal{L}}}{\partial \mu}$
$W \leftarrow W + \frac{\partial \hat{\mathcal{L}}}{\partial W}$ and $\mu \leftarrow \mu + \frac{\partial \hat{\mathcal{L}}}{\partial \mu}$, updating network weights
end while
return trained network $\hat{\Lambda}$
Stop

4.6.1 Backprobagation Algorithm for Performance Ratio Maximization

To derive with respect to the weights of the network, we need to calculate the derivative of a function G with respect to the outputs of the network which are the portfolio decisions x_i ($\partial G/\partial x_i$ for $i = 1, \ldots, N$). The performance measures are all fraction where a reward function is divided by a specific risk function. Suppose $\psi = \frac{\phi}{\rho}$ is a function of a performance ratio for the period j where the index j is removed for simplicity. Therefore, to compute the following partial derivative we have:

$$\frac{\partial \psi}{\partial x_i} = \frac{\frac{\partial \phi}{\partial x_i}\rho - \frac{\partial \rho}{\partial x_i}\phi}{\rho^2} \tag{4.20}$$

For simplicity, consider $\partial \psi/\partial x_i = \psi'_i$, $\partial \phi/\partial x_i = \phi'_i$, and $\partial \rho/\partial x_i = \rho'_i$. In what follows, two last derivatives are obtained for all performance measures for one sample period. For this, we estimate all the expectations based on finite observations with equal probabilities.

In all ratios except the Rachev ratio, the reward function is the expected value of

average returns during a period defined by the following equation:

$$\begin{aligned} E\left[r_p\right] &= \frac{1}{D}\sum_{t=1}^{D}\sum_{i=1}^{N} x_i r_{it} \\ &= \sum_{i=1}^{N} x_i \bar{r}_i \end{aligned} \tag{4.21}$$

Then, $\phi_i' = \bar{r}_i$. Here, $\bar{r}_i$ is the sample average return of ith asset in the portfolio, and D denotes the number of working days in this period. $(x_1, \ldots, x_N)$ is the vector of portfolio decisions for N assets.

Sharpe

We estimate the variance for samples based on the following formula:

$$\hat{\sigma} = \frac{\sum_{i=1}^{n}(X_i - X_{avg})^2}{n-1} \tag{4.22}$$

Here, n the number of samples, X_i is the i^{th} sample, and X_{avg} indicates the average of samples. Consider also $\hat{\sigma}_{i,j} = \text{cov}\left(i,j\right)$. The variance of a portfolio is calculated by the definition of the standard deviation of the sum of variables:

$$\begin{aligned} \hat{\sigma}_{r_p} &= \left(\text{var}\left(\sum_{i=1}^{N} x_i r_i\right)\right)^{1/2} \\ &= \left(\sum_{i=1}^{N} x_i^2 \text{var}\left(r_i\right) + 2\sum_{i}\sum_{j>i} x_i x_j \text{cov}\left(r_i, r_j\right)\right)^{1/2} \\ &= \left(\sum_{i=1}^{N} x_i^2 \hat{\sigma}_{r_i}^2 + 2\sum_{i}\sum_{j>i} x_i x_j \hat{\sigma}_{r_i, r_j}\right)^{1/2} \end{aligned} \tag{4.23}$$

Using the Equation (4.23), we can compute the partial derivative for the risk as

follows:

$$\rho_i' = \frac{x_i \hat{\sigma}_{r_i}^2 + \sum_{j \neq i} x_j \hat{\sigma}_{r_i, r_j}}{\left(\sum_{i=1}^{N} x_i^2 \hat{\sigma}_{r_i}^2 + 2 \sum_i \sum_{j>i} x_i x_j \hat{\sigma}_{r_i, r_j} \right)^{1/2}} \tag{4.24}$$

MAD

The estimation of the risk in MAD based on samples is:

$$\begin{aligned} \rho &= E\left[\left| \sum_{i=1}^{N} x_i r_i - E\left[\sum_{i=1}^{N} x_i r_i \right] \right| \right] \\ &= \frac{1}{D} \sum_{t=1}^{D} \left| \sum_{i=1}^{N} x_i (r_{it} - \bar{r}_i) \right| \end{aligned} \tag{4.25}$$

consider $d_{it} = r_{it} - \bar{r}_i$ for asset i at time t. Then, it is simplified as:

$$\begin{aligned} \rho =& \frac{1}{D} \sum_{t=1}^{D} \left| \sum_{i=1}^{N} x_i d_{it} \right| \\ =& \frac{1}{D} \sum_{t=1}^{D} \max\left(\sum_{i=1}^{N} x_i d_{it}, -\sum_{i=1}^{N} x_i d_{it} \right) \end{aligned} \tag{4.26}$$

Then,

$$\rho_i' = \frac{1}{D} \sum_{t=1}^{D} \left(d_{it}.1_{(x_i d_{it} \geq 0)} - d_{it}.1_{(x_i d_{it} < 0)} \right) \tag{4.27}$$

Here, $1_{(.)}$ is an indicator function. If the condition in the parenthesis is satisfied it is equal to 1, otherwise it is zero.

MiniMax

When a decision vector is fixed and the portfolio returns are released in D days, we sort them ascendingly and let $r_p(1)$ denote the smallest value for portfolio return during a period. Therefore, we have:

$$\rho_i' = r_{it}\,, \;\; \{t : r_{p_t} = r_p(1)\} \tag{4.28}$$

Gini

If there is no distribution assumption for the portfolio return, the following expression for Gini measure is given for discrete case (Yitzhaki, 1982):

$$\begin{aligned}\Gamma_p &= \frac{1}{D(D-1)} \sum_{t=1}^{D-1} \sum_{t'>t}^{D} |r_{p_t} - r_{p_{t'}}| \\ &= \frac{1}{D(D-1)} \sum_{t=1}^{D-1} \sum_{t'>t}^{D} \left| \sum_{i=1}^{N} x_i (r_{it} - r_{it'}) \right|\end{aligned} \tag{4.29}$$

In the Equation (4.29), let $\delta_{itt'} = r_{it} - r_{it'}$. We can rewrite it as follows and estimate the Gini risk:

$$\rho = \frac{1}{D(D-1)} \sum_{t=1}^{D-1} \sum_{t'>t}^{D} \left| \sum_{i=1}^{N} x_i \delta_{itt'} \right| \tag{4.30}$$

Similar to the derivative for MAD, we have:

$$\rho_i' = \frac{1}{D(D-1)} \sum_{t=1}^{D-1} \sum_{t'>t}^{D} (\delta_{itt'}.1_{(x_i\delta_{itt'} \geq 0)} - \delta_{itt'}.1_{(x_i\delta_{itt'} < 0)}) \tag{4.31}$$

CVaR

First, we need to determine the VaR at a given level α and then, calculate the expectation of the values which are less than VaR_α. When a decision is fixed, we observe the portfolio returns for a period including D days. The α-quantile of discrete values is the return value at point $V_\alpha = \lceil D\alpha \rceil$ of increasingly sorted return values where $\lceil a \rceil$ is the smallest integer number which is equal or greater than a. CVaR_α is the expectation of sorted values which are below the VaR_α, hence:

$$\rho = \text{CVaR}_\alpha = \frac{1}{V_\alpha} \sum_{j=1}^{V_\alpha} -r_p(j) \tag{4.32}$$

Here, $r_p(j)$ is the jth smallest value. The derivative of the above function with respect

to the weight i is:

$$\rho_i' = \frac{1}{V_\alpha} \sum_t -r_{it} \ , \ \forall \, t : r_{p_t} = r_p(j) \ , \ j = 1 : V_\alpha \tag{4.33}$$

Rachev

In the Rachev ratio, in addition to the risk, the reward is defined by CVaR at a given level of probability, β. Then, the derivative of the risk is similar to the CVaR ratio, and the reward and its derivative with respect to portfolio weight i are:

$$\phi = \frac{1}{D - V_\beta + 1} \sum_{j=V_\beta}^{D} r_p(j) \tag{4.34}$$

$$\phi_i' = \frac{1}{D - V_\beta + 1} \sum_t r_{it} \ , \ \forall \, t : r_{p_t} = r_p(j) \ , \ j = V_\beta : D \tag{4.35}$$

4.6.2 DNN Model Tuning

In practice, cross-validation is implemented to prevent the model from overfitting and underfitting. We divide the whole sample into different windows. Each window is partitioned into train and test periods and each training period includes a smaller train set and a validation set. Deep neural networks are trained with different hyper-parameters using the small train set in each window. This includes the size of the network, initial random weights, and learning rate to reach the best optimal point of the objective function. The network structure and the number of hidden layers and nodes are specified by the formulas introduced in the literature and explained in Chapter 2. The initial weights of the network are generated by random sampling from a standard normal distribution, $w \in \mathcal{N}(0, 1)$. The learning rate or step size is considered as a step decaying at each iteration of the BP algorithm and computed as $\gamma_i = \gamma_0/(1 + ci)$. Here, γ_i is the step size of i^{th} iteration, and γ_0 and c are constants. We do a grid search over different values of γ_0 and c. To select the best hyperparameters, we compare values of function and pick those corresponding to the highest

value. To choose the best performance ratio in Section 4.11.1 and train the network for classification, we build the model with a various number of hidden layers and a different number of hidden nodes in each layer.

4.6.3 Portfolio Choice in a New Period

So far, we have shown how to train DNN and find the optimal weights of assets in a portfolio that maximize a specific performance ratio. To determine optimal weights in the new period $T+1$ ($\mathbf{x}_{T+1}$), we use the trained DNN of the ratio for that period and the state variables to find the optimal portfolio weights. More formally,

$$\mathbf{x}_{T+1} = \hat{\Lambda}(\mathbf{z}_{T+1}) \tag{4.36}$$

4.7 Benchmarks

In this study, we use the first-order Vector Autoregressive (VAR) model (Bekaert and Hodrick, 1992) and factor model as two-step methods for the benchmark model to estimate the mean vector and covariance matrix. In the first method, moments of returns are fitted into a Vector Autoregressive (VAR) model over the in-sample periods, and then slopes are used to estimate moments and maximize performance ratios. The factor model is based on the four state variables as factors in this study, and the following conditional terms show this:

$$\Sigma = \mathrm{Var}[r|Z] \qquad \mu = E[r|Z] \tag{4.37}$$

Here, Σ is the covariance matrix. The mean vector and covariance matrix are estimated by linear regression on the state variables. We consider these parameter estimations in a two-step classical approach as the benchmark to which our method, suggested in the following section, is compared. The main disadvantage of the classical approach is that it requires the assumption of a particular form for the shape of

the returns, however, the empirical returns distributions are rarely known or follow any regular distribution. Moreover, this two-step optimization leads to a combination of parameter estimation error and model optimization error.

We also consider a more competitive benchmark method that integrates estimation and optimization steps. In this method, a parametric method proposed by Brandt et al. (2009) is used to estimate optimal weights directly from the state variables. In this benchmark, an expected utility function is maximized during the in-sample period to obtain the coefficients in a linear model that relates state variables to portfolio weights. The details and specifics of these three benchmark methods that we use in this study are as follows.

VAR-model: first, we calculate monthly moments (mean vector and covariance matrix) of assets using daily returns over the in-sample periods, and then, these monthly moments are fitted into an autoregressive process with one lag (AR (1)) as a VAR model over the in-sample periods. During the out-of-sample periods, the estimated slopes and the realized moments (as input) are used to estimate monthly moments. We use the monthly estimated moments to simulate daily returns of the assets and consequently maximize the performance ratios to determine the optimal weights per month and during the out-of-sample period.

Factor-model: as a first step, we calculate monthly moments (mean vector and covariance matrix) of assets using daily returns over the in-sample periods. Then, we use a factor model that includes the four state variables that we have used in this paper (i.e., the default spread, the dividend yield, the term spread, and the S&P 500 Index trend) and regress monthly moments on the set of state variables during the in-sample period. During the out-of-sample periods, the estimated slopes and state variables (as input) are used to estimate monthly moments. We use the monthly estimated moments to simulate daily returns of the assets and consequently maximize the performance ratios to determine the optimal weights per month and during the out-of-sample period.

Brandt-model: following Brandt et al. (2009), we consider the Constant Relative

Risk Aversion (CRRA) utility function with a prespecified risk parameter, $\gamma = 5$. We then optimize the expected utility function over in-sample periods to obtain the optimal coefficients in a linear model that makes the portfolio weights dependent on the state variables. In out-of-sample periods, we use the estimated linear model and the new state variables at the beginning of each period to find the optimal portfolio weights per month and during the out-of-sample period.

4.8 Data

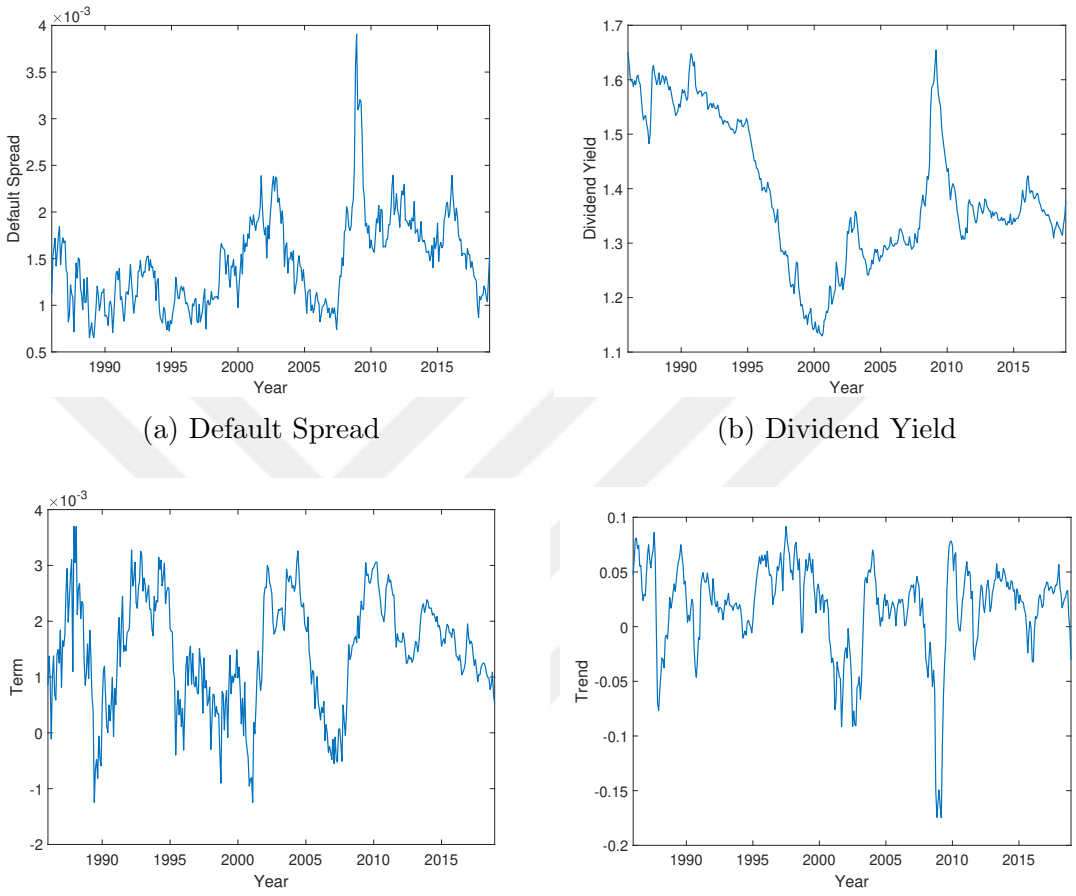
We use real-world data to implement our approach and to examine the impact of market states in portfolio optimization. We construct a daily portfolio rebalanced monthly of the S&P 500 index and a Bond index which is a US benchmark thirty-year government index, from January 1986 to December 2018, sourced from Thomson Reuters. We follow the intuition of Aït-Sahalia and Brandt (2001) and select the default spread, the log dividend-to-price ratio of the S&P 500 index, the term spread, and the S&P 500 index trend monthly as our state variables. Monthly data on S&P 500 is collected from Robert J. Shiller's website at `http://www.econ.yale.edu/~shiller/`, accessed on 14/6/2019.

We define the Default Spread as the monthly difference between Moody's Baa- and Aaa-rated Corporate Bond Yields, Dividend Yield is log dividend-to-price ratio, Term Spread is the monthly difference between the rates on 10- and one-year Treasury Constant Maturity Rates, and Trend is the log of the ratio of the current S&P index level and the average index level over the previous 12 months. The total daily observations of S&P 500 and bond are $d = 8609$ data points. The monthly data of state variables contains $T = 396$ observations.

Figure 4.2 depicts the time series for state variables. The period of study includes a couple of economic and financial events. The first event dates back to 1987 as the global stock markets crashed including in the U.S., when S&P 500 index lost 58 points or 20.47% of its value on October 19, known as Black Monday which marked the beginning of a global stock market decline. Afterward, the Dotcom bubble which

was a sector-based crisis occurred in 1999-2000. It was followed by a bull rush into technology and internet-related stocks. Individuals became rich rapidly through companies such as eBay and Amazon. In 2003, the U.S involvement in the war in Iraq imposed $944 billion for operations overseas, and this remarkable financing had long-lasting effects on the economy. The first signs of the Global Financial Crisis (GFC) started in 2007 with the sub-prime mortgage crisis. The housing prices began falling and caused the Federal Reserve to add $24 billion in liquidity to the banking system. It raised interest rates which made households pay more for their adjustable-rate mortgages. The stock market crashed in August 2008 and market industrial indices declined drastically. Figure 4.2(b) shows the formation of a bubble in dividends and bursting before the crash in the market. In addition, Figure 4.2(d) denotes the downturn through the negative difference between the S&P level and the average of 12 previous values indicating the economy shrinkage of 0.3 percent.

Table 4.1 shows the summary statistics of indices and state variables. Panel A shows daily statistics for S&P 500 and Bond index returns and the correlations between them, as well as daily statistics for Treasury Bill rates which refer to as risk-free benchmark in this study. Panel B represents monthly statistics and correlations between state variables. Panel C reports the Jarque-Bera normality test on daily returns of indices. The S&P index shows a higher average and standard deviation than those of the Bond index. This difference reflects the theory that a higher return is achievable by taking on more risk. Both of the indices have a small but negative skewness. Kurtosis for the S&P index suggests that it is very fat-tailed compared to the normal distribution, while the bond index has smaller kurtosis over a value of 3. The correlation between the two indices is relatively small and negative. Amongst correlations of state variables, the trend has negative correlations with all other variables. The default spread and the trend are highly negatively correlated, and the log dividend-to-price ratio and the trend show a smaller negative correlation. The term has small positive and almost equal correlations with default spread and log dividend-to-price ratio. The correlation between default spread and log dividend-to-price ratio



(a) Default Spread (b) Dividend Yield

(c) Term Spread (d) Trend

Figure 4.2: Monthly time series of four state variables. The sample is from January 1986 to December 2018.

is negligible however negative.

The main purpose of considering various performance ratios is to take into account different moments and shapes of the index return distribution. The classical definition of risk refers to using standard deviation as the dispersion measure in normal data. Since this parameter does not perfectly introduce the risk for non-normal returns, recent ratios are based on the importance of fat tail returns which is ignored in classical definition. Therefore, we implement a test to see whether the returns of indices follow a normal distribution. For this, the Jarque–Bera (JB) normality test is used, as proposed by Jarque and Bera (1987). The null hypothesis is the normal

distribution of the data. The test statistic has a chi-square distribution for large sample sizes. Panel C of table 4.1 shows the result of this test on daily S&P and Bond indices returns. We reject the null hypothesis that returns are normally distributed at the confidence level of 1%.

Table 4.1: Descriptive statistics, correlations for indices and state variables (reported in percent), and normality test on indices.

Panel A: Indices and Benchmark (Daily)

	Mean	Median	StdDev	Skewness	Kurtosis	Correlations S&P	Correlations Bond
S&P	0.09	0.08	0.18	-0.84	24.94	1.00	-0.13
Bond	0.08	0.06	0.13	-0.03	6.17		1.00

Panel B: State Variables (Monthly)

						Def	LnDP	Term	Trend
Def	0.15	0.14	0.05	1.06	5.42	1.00	-0.09	0.22	-0.56
LnDP	1.39	1.36	0.13	0.16	2.06		1.00	0.25	-0.12
Term	0.15	0.15	0.10	-0.30	2.52			1.00	-0.14
Trend	1.81	2.49	4.18	-1.65	7.27				1.00

Panel C: Jarque–Bera test on daily returns

S&P			Bond			critval (1%)
h	p-value	JBstat (χ^2_2)	h	p-value	JBstat (χ^2_2)	11.03
1	0.00	170.96	1	0.00	584.66	

4.9 Empirical Application

We construct portfolios using the Standard and Poor (S&P) 500 index and the US benchmark thirty-year government index. Daily returns on these indices are sourced from Refinitiv Thomson Reuters and cover the period from the beginning of January 1986 to the end of December 2018.

4.9.1 DNN versus Traditional Optimal Portfolio Choice

As the objective function in a portfolio choice, various performance ratios are discussed in the literature. In this study, we choose six popular performance measures to confirm the outperformance of DNN compared to traditional methods. However, our machine learning procedure is not restricted to the type of performance measures. We use Sharpe, Mean Absolute Deviation (MAD), MiniMax, Gini, CVaR, and Rachev ratios. These performance measures are widely used in the literature and include both symmetric and asymmetric measures.

We use rolling 156-month (13-year) observations from January 1986 as a training set (or in-sample) and test the trained model on the following 60 months as a test set (or out-of-sample). We roll over the training and test sets every 60 months. Figure 4.3 shows the rolling windows of training and test sets. This figure shows how the entire sample period of January 1986 to December 2018 is partitioned into training (in-sample) and test (out-of-sample) samples. We use 13 years of data starting from January 1986 as the training set and the following five years as the test set. We roll over these sets for five years until the end of the sample in December 2018. This approach generates four training sets and four consecutive test sets of 1999-2003, 2004-2008, 2009-2013, and 2014-2018.

This approach provides four 5-year consecutive test periods starting in January 1999 until the end of our sample in December 2018. Test periods are 1999–2003, 2004–2008, 2009–2013, and 2014–2018. For each test period, DNN is trained over the previous 156 months of observations. We did not break our sample simply into two sub-samples of training and test, as the rolling windows allow us to include the

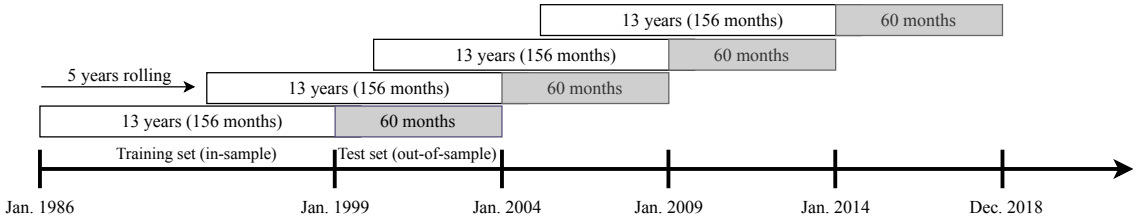


Figure 4.3: Sample period partitioning and rolling windows.

most recent information in the predicting model. We did not roll the windows more frequently than every five years as a shorter frequency is unlikely to provide more information to the trained model, given we use monthly state variables.

We apply our designed DNN to construct monthly portfolios given the state of the market at the beginning of the month. As the first step, we use state variables at the beginning of each month and daily returns of the stock and bond indices during the month, over the training sets, and train a DNN. These trained DNNs are predicting models (for each performance ratio) that take the state variables at the beginning of the month as the input and give the optimal weights for that month as the output. These are the weights that maximize the associated ratio for a given month, conditioned on the market state. It is important to investigate if using DNN in the maximization of the ratios outperforms the traditional methods.

We employ two traditional methods that estimate the moments of returns and maximize the ratios: a VAR and a factor model. The VAR model is an Autoregressive Process with one lag (AR (1) process) and the factor model includes the four state variables that we have used in this chapter. The former is unconditional, whereas the latter is a conditional approach. In both methods, we use the models to predict the moments of returns over the in-sample periods, first and then maximize the performance ratios during the test periods (total of 240 months). We also use a more competitive benchmark suggested by Brandt et al. (2009) as the third benchmark, where a utility function is maximized to obtain the portfolio choice. In this model, the portfolio weights are linearly dependent on the state variables, then decision vari-

ables are linear coefficients associated with each state variable. In this study, similar to Brandt et al. (2009), we consider the Constant Relative Risk Aversion (CRRA) utility function with a parameter. We then optimize the expected utility function over in-sample periods to obtain the optimal linear coefficients and consequently optimal portfolio weights. In out-of-sample periods, we use the linear portfolio policy and the new state variables at the beginning of each period during the test to find the optimal portfolio weights. The objective function is the expected utility that would implicitly take into account the different moments of the return distribution and the risk profile of the investor. To compare the performance of this benchmark with the DNN method, we use the obtained optimal portfolio weights and compute the realized portfolio return out of the sample and calculate the performance of the portfolios based on different ratios considered in this study.

We also use trained DNNs over the same 240-month test period. Table 4.2 shows the descriptive statistics of the performance ratios maximized using DNN as well as three benchmark methods: a Factor model and a VAR model. The VAR model is an autoregressive process with one lag (AR (1)) and the Factor model includes Default Spread, Term Spread, Dividend Yield, and Trend. For the third benchmark, we maximize the utility function with a risk parameter during similar out-of-sample periods which is a total of 240 months. These methods are used for the four test sample periods (1999-2003, 2004-2008, 2009-2013, and 2014-2018) to determine optimal portfolio weights and compute the predicted value of ratios for each month. For each test sample period and each ratio, the prior 13 years' data are used to train DNN and estimate the predicting parameters of the factor and VAR models. The table presents the summary statistics of the monthly predicted value of performance ratios over the whole period of 1999-2018. It also shows the significance level of the mean difference between values of a ratio using DNN and the traditional method that generates a higher value (i.e., the best benchmark).

The means of all ratios are greater when we use DNN to find the optimal weights compared to other methods. For example, using the DNN-based approach, the aver-

age Sharpe ratio is 0.09 on monthly basis, whereas it is 0.08 based on the benchmark method Brandt, 0.07 based on the benchmark method factor, and 0.06 based on the benchmark method VAR. The biggest improvement is for CVaR and MAD ratios with around 50 % increase in the values with respect to the benchmark method Brandt. For Rachev, there is an increase from 3.44 under the benchmark method Brandt to 4.45 using the DNN-based approach, and the mean difference between the two is significant at the 1 % level.

Additionally, of the three benchmarks, the greater ratio value is obtained by the benchmark method Brandt, which is an integrated linear model. This result supports our model that integrating estimation and optimization steps alleviates the errors that occur in the estimation step and improves the objective function. The second-best method is the benchmark factor method, a conditional approach, compared to the unconditional benchmark VAR method, except for the MiniMax ratio. These results show the importance of considering the state of the market in the asset allocation problem as well as the integration of estimation and optimization steps. However, a factor model used in the benchmark factor method and the integrated version in benchmark method Brandt assume that the relationship between return moments and market state variables is linear. Our results show that DNN provides a better outcome as it captures the nonlinear relations between these variables and the portfolio weights using an integrated approach. In other words, improvement for all ratios using the DNN-based approach compared to the benchmarks is not only due to considering the state of the market in portfolio choice. Table 4.2 also reports other statistics of each performance ratio. Standard deviations of values of ratios obtained by ANN are relatively close to benchmarks, while the negative skewness disappears for Sharpe, CVaR, and Rachev. Skewness and kurtosis increase using ANN compared to benchmark methods factor and VAR for all ratios except for Rachev, for which these values are close. While skewness and kurtosis obtained by benchmark method Brandt are larger than ANN method in all cases except skewness in CVaR and Rachev.

To demonstrate the performance of DNN compared to benchmark methods over

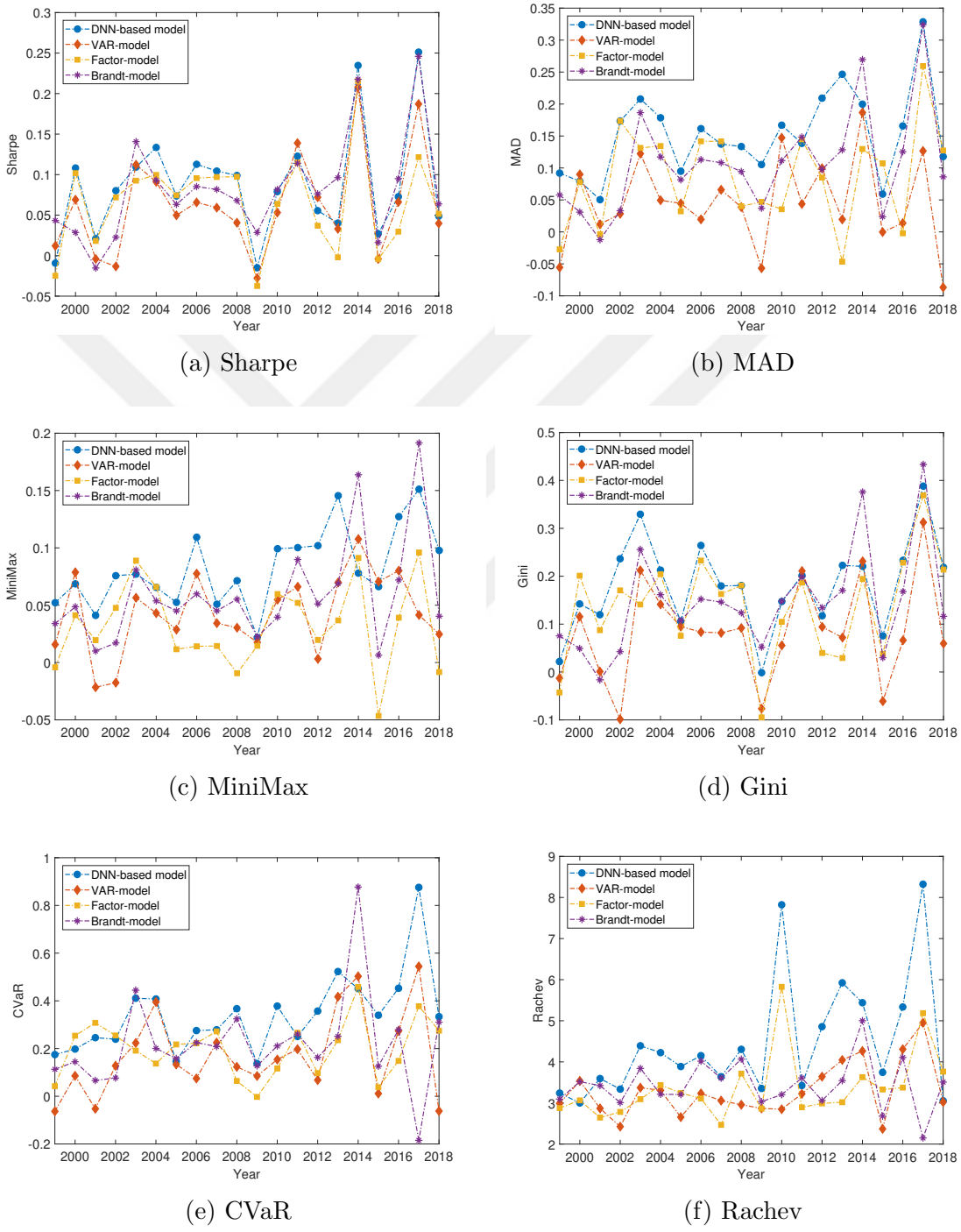
Table 4.2: Performance ratios: DNN versus traditional estimates of the portfolio choice.

	Mean	StdDev	Skewness	Kurtosis
Sharpe				
VAR-model	0.06	0.20	−0.05	2.88
Factor-model	0.07	0.20	−0.10	2.73
Brandt-model	0.08	0.20	0.38	3.77
DNN	0.09	0.20	0.12	3.28
MAD				
VAR-model	0.05	0.24	0.07	2.80
Factor-model	0.09	0.26	0.05	2.95
Brandt-model	0.10	0.25	0.35	3.66
DNN	0.15***	0.26	0.06	3.35
MiniMax				
VAR-model	0.04	0.12	1.27	7.58
Factor-model	0.03	0.12	0.91	5.22
Brandt-model	0.06	0.14	2.98	22.42
DNN	0.08***	0.14	1.43	6.32
Gini				
VAR-model	0.08	0.35	0.08	2.74
Factor-model	0.14	0.35	0.01	2.79
Brandt-model	0.14	0.35	0.35	3.61
DNN	0.18***	0.34	−0.01	3.43
CVaR				
VAR-model	0.17	0.51	3.22	9.24
Factor-model	0.20	0.477	1.98	8.13
Brandt-model	0.22	0.70	−2.47	48.86
DNN	0.34**	0.64	3.48	13.26
Rachev				
VAR-model	3.33	2.01	2.59	14.48
Factor-model	3.36	3.10	8.32	97.88
Brandt-model	3.44	2.69	−1.25	31.07
DNN	4.45***	4.51	5.26	40.01

Note. Significance levels refer to the t-test for mean difference between DNN and the best of benchmarks. $^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

time, for each year we compute the monthly average of the predicted values of each ratio using different methods. We show the time series of means in Figure 4.4. The plots

show that almost in all years and across all ratios DNN provides better predictability power.



(a) Sharpe (b) MAD

(c) MiniMax (d) Gini

(e) CVaR (f) Rachev

Figure 4.4: DNN-based versus traditional estimates of portfolio choice. This figure displays the time series of performance ratios using the proposed DNN-based approach as well as three traditional methods as benchmarks.

We further present the performance of DNN during sub-samples of the whole test period i.e. 1999–2003, 2004–2008, 2009–2013, and 2014–2018 in Table 4.3. These periods are associated with important economic and financial turbulence. The 1999–2003 period includes the Dotcom bubble, 2004–2008 started with a bull market and ended with the Global Financial Crisis (GFC), and 2009–2013 is associated with the post-GFC period. Finally, 2014–2018 includes the recent growth period. The results from subsamples confirm our findings in Table 4.2. This table shows the mean value of the performance ratios maximized during the test sample periods of 1999-2003, 2004-2008, 2009-2013, and 2014-2018 using DNN as well as two benchmark methods: a Factor model and a VAR model. The VAR model is an autoregressive process with one lag (AR (1)) and the Factor model includes Default Spread, Term Spread, Dividend Yield, and Trend. These methods are used for the four test sample periods to determine optimal portfolio weights and compute the predicted value of ratios for each month. For each test sample period and each ratio, the prior 13 years' data are used to train DNN and estimate the predicting parameters of the factor and VAR models.

The average of ratios is higher when we use DNN compared to the factor model, VAR, and Brandt methods. In almost all subsamples and for all ratios, the difference between the mean values of the ratios using DNN and the best benchmark is positive and significant. This suggests that DNN is robust in outperforming other traditional methods in different episodes of the market.

Static Benchmark

The static approach is a naïve model that is used as a simple benchmark as Benchmark-static$_{\text{x-y}}$, where x and y denote the weights of the S&P 500 index and the bond index in the portfolios, respectively. In this model, we consider three types of such static portfolios; 1) Benchmark-static$_{80\text{-}20}$: the weight of the SP 500 index is 80 %, and 20 % in the bond index, 2) Benchmark-static$_{60\text{-}40}$: the weight of S&P 500 index is 60 %, and 40 % in the bond index, 3) Benchmark-static$_{20\text{-}80}$: the weight of S&P 500 index

is 20 %, and 80 % in the bond index. We use these fixed weights to obtain portfolio returns during out-of-sample periods. We then use performance ratios, to calculate the portfolio performance with each ratio.

Benchmark-static$_{20\text{-}80}$: Sharpe=0.05, MAD=0.06, MiniMax=0.04, Gini=0.09, CVaR=0.18, Rachev=3.31,

Benchmark-static$_{60\text{-}40}$: Sharpe=0.07, MAD=0.09, MiniMax=0.05, Gini=0.12, CVaR=0.25, Rachev=3.56,

Benchmark-static$_{80\text{-}20}$: Sharpe=0.07, MAD=0.09, MiniMax=0.06, Gini=0.12, CVaR=0.17, Rachev=3.53.

In this study, we have used four state variables as the determinants of the optimal portfolio weights. It is interesting to examine the relative importance of each state variable in the predicted values of performance ratios. Since DNN uses a non-linear approach we do not simply regress time series values of the predicted ratios on the lagged state variables. In the next section, we apply two methods to understand better the impact of the state variables on the performance ratios. For brevity, we report and discuss this impact only on the Sharpe ratio, as the most common performance ratio used by practitioners and academics in Finance, and only present the plots for other ratios.

Weight Analysis

We analyze the portfolio weights constructed using the Sharpe ratio for all the methods and compute the average weight of the S&P 500 index and the average turnover in each year. Figure 4.5 shows the plots for our method and the benchmarks. Portfolio turnover is defined as the average changes in portfolio weights over consecutive periods (DeMiguel et al., 2009). If we denote the portfolio weight for asset i chosen at time t by $x_{i,t}$, the portfolio turnover for N assets over periods $t = 0, \ldots, T-1$ is:

$$Turnover = \frac{1}{T-1} \sum_{t=0}^{T-1} \sum_{i=1}^{N} |x_{i,t+1} - x_{i,t}| \tag{4.38}$$

Table 4.3: Performance Ratios: DNN versus traditional estimates of the portfolio choice in different subsamples.

	1999-2003	2004-2008	2009-2013	2014-2018
Sharpe				
VAR-model	0.04	0.06	0.05	0.10
Factor-model	0.05	0.09	0.04	0.08
Brandt-model	0.04	0.08	0.08	0.12
DNN	0.06***	0.10	0.06	0.13
MAD				
VAR-model	0.04	0.04	0.05	0.05
Factor-model	0.07	0.10	0.06	0.12
Brandt-model	0.06	0.10	0.10	0.16
DNN	0.12*	0.14*	0.17**	0.17
MiniMax				
VAR-model	0.02	0.04	0.04	0.06
Factor-model	0.04	0.02	0.04	0.03
Brandt-model	0.04	0.05	0.05	0.09
DNN	0.06***	0.07*	0.09**	0.10
Gini				
VAR-model	0.04	0.10	0.07	0.12
Factor-model	0.11	0.17	0.05	0.21
Brandt-model	0.08	0.13	0.14	0.22
DNN	0.17*	0.19	0.14	0.23
CVaR				
VAR-model	0.06	0.19	0.18	0.25
Factor-model	0.21	0.18	0.14	0.26
Brandt-model	0.17	0.22	0.20	0.28
DNN	0.25	0.30*	0.33**	0.49
Rachev				
VAR-model	3.04	3.04	3.32	3.78
Factor-model	2.89	3.19	3.52	3.86
Brandt-model	3.37	3.62	2.29	3.49
DNN	3.51	4.04*	5.08***	5.18**

Note. Significance levels refer to the t-test for mean difference between DNN and the best of benchmarks. $^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

It is observed that the average weight on equity index and portfolio turnover in both DNN and benchmark-Brandt are similar and quite low. However, these plots are similar for benchmark-var and factor models with high turnover.

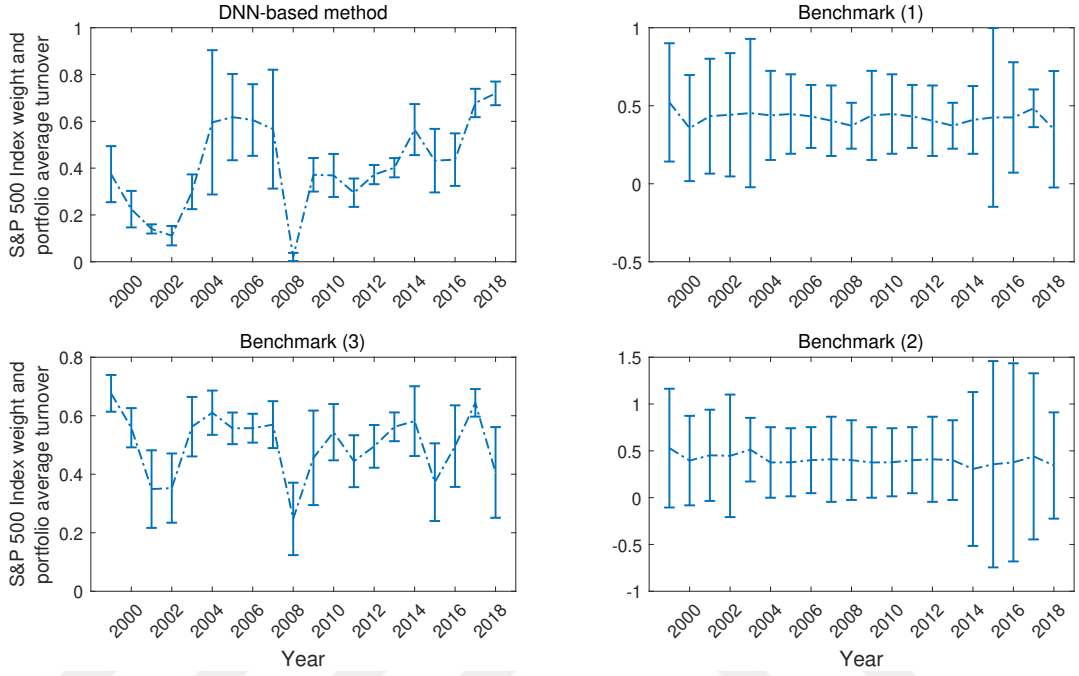


Figure 4.5: Weight of S&P 500 index and portfolio turnover using Sharpe ratio.

4.9.2 Sharpe Ratio and State Variables

The use of new statistical methods like DNN improves the prediction and estimation in many different problems. This advantage is obtained by the complexity of these methods and their ability in nonlinear modeling (Olden and Jackson, 2002). As a result, neural networks such as DNN are considered as "Black Box" with excessive connection weights which provides little insight into the influence of input variables on the response variable. However, interpretation of statistical models is desired in many fields such as Finance where it gives the practitioners the knowledge on relationships leading to behavior detection.

Several methods have been introduced in the literature which identify the importance of explanatory variables and their relative contributions to the modeling process in neural networks. We use a method called "Connection Weights" to find the relative importance of the state variables in predicting the Sharpe ratio. This

method has been shown the most successful results to determine the importance of independent variables in a network with one hidden layer and one output node (Olden and Jackson, 2002). In this approach, the estimated values of network weights are used to estimate the relative importance (RI) measure as follows.

Recall the network in Figure 4.1 which has M input nodes, H hidden nodes, and one output node $o = 1$. Let w_{ih} and w_{ho} indicate the network weighs between state variable i and hidden node h, and the network weight between hidden node h and output node o (portfolio weights used to estimate Sharpe ratio), respectively. For state variable i, we calculate the RI measure as:

$$RI_i = \sum_{h=1}^{H} w_{ih} \cdot w_{ho} \quad \forall\, i = 1, \cdots, M \tag{4.39}$$

The larger (smaller) values of RI indicate the higher (lower) importance of the associated state variable in determining portfolio weights and consequently estimated Sharpe ratio. We use the estimated network weights of the predicting DNNs for each subsample and compute the RI measure for each state variable.

Table 4.4: Relative importance of the state variables as determinants of Sharpe ratio.

	Average	1999-2003	2004-2008	2009-2013	2014-2018
Default Spread	3.4	1.6	4.6	2.3	4.9
Dividend Yield	2.3	1.0	4.6	2.9	0.5
Term Spread	1.0	1.5	0.7	0.3	1.3
Trend	5.3	5.9	8.4	5.1	1.8

Table 4.4 reports the results. The trend is the most important state variable to predict Sharpe ratio across all subsamples except for the sample period of 2014-2018 during which Default Spread is the most important one followed by Trend. Over the sample periods of 1999-2003 and 2009-2013, Default Spread and Dividend Yield are the second most important variables, respectively. During the sample period of 2004-2008, both Default Spread and Dividend Yield have equal RIs suggesting they

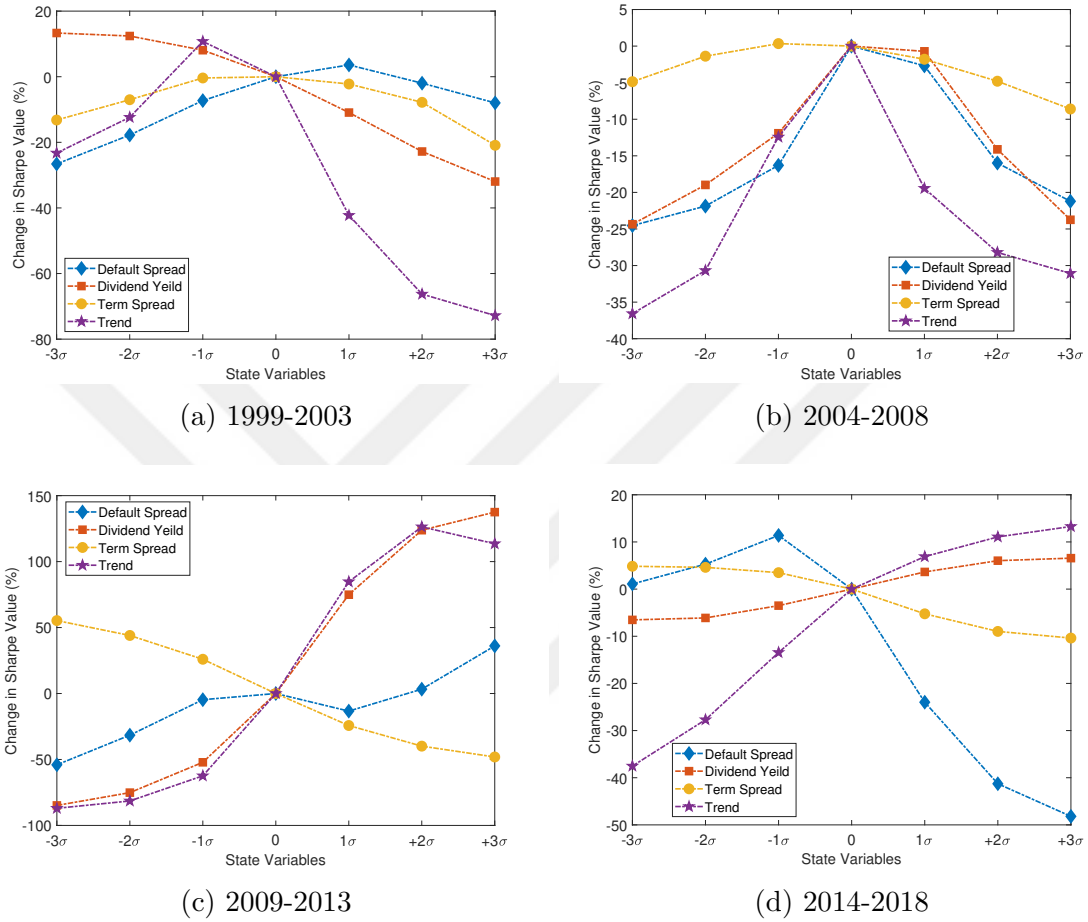
have equal importance in predicting the Sharpe ratio. In summary Table, 4.4 shows that Trend (Term Spread) is the most (least) influential state variable in estimating Sharpe ratio, followed by Default Spread and Dividend Yield.

The relative importance approach does not give us any indication about the direction and magnitude of the impact of variables on the Sharpe ratio. To do so, we conduct a sensitivity analysis called the "Perturb" method which is an ideal approach to examine the direction and significance of the impact of an independent variable on the output in a complex, non-linear model with various independent variables.

The sensitivity analysis is a visual presentation showing the impact of small changes in each parameter of a model on the value of the function (Sharpe ratio) in question, all on the same graph. In our analysis, values of a state variable, as the input of the predicting DNN, are varied across a range of values and the effect on the Sharpe ratio is observed in each scenario. The values of other state variables are the actual data and remain unchanged in different scenarios. Then we display the range of changes in the Sharpe ratio against the changes in each state variable. In our analysis, we change the values of a state variable by $\pm 3\sigma$, $\pm 2\sigma$, and $\pm\sigma$, where σ is its standard deviation, and calculate the Sharpe ratio using the predicting DNN over the test subsamples. We plot the percent change in the estimated Sharpe ratio against these variations for each state variable and subsample, as shown in Figure 4.6.

The Y-axis shows the expected range of Sharpe ratio if the value of either Trend, Default Spread, Term, or Dividend Yield is shifted somewhere between a minimum and maximum of the fixed range we have defined based on their deviations from their means (standard deviations). The plots on this figure show how the Sharpe ratio increases or decreases as the values of one state variable are perturbed and all others held on their original values. The plots reveal the non-linear relationships and the relative sensitivity of the Sharpe ratio to changes in each state variable.

Figure 4.6(a) shows the plot for the 1999-2003 period which includes the Dot-com crash that lasted from March 2000 to October 2001. The largest Sharpe ratio



(a) 1999-2003

(b) 2004-2008

(c) 2009-2013

(d) 2014-2018

Figure 4.6: Sensitivity analysis for state variables against Sharpe ratio in a DNN network.

range is for Trend, followed by Dividend Yield, and changes in all state variables have a negative impact on Sharpe ratio. The exception is Dividend Yield for which its negative change has a positive impact on Sharpe ratio.

The plot for Trend and Dividend Yield also reveals that positive change in Trend has a more pronounced negative effect on Sharpe ratio than positive change in Dividend Yield. The direction of the effect is opposite for the extreme negative changes in Trend and Dividend Yield. The extreme negative changes in Trend (Dividend Yield) decrease (increase) the Sharpe ratio. Default Spread and Term Spread have a close and negative impact on Sharpe ratio.

The plots from the period 2004-2008 (Figure 4.6(b)), which includes GFC, provide almost similar, but stronger results. Sharpe ratio is more sensitive to Trend, followed by Dividend Yield and Default Spread. Furthermore, any positive or negative changes in any state variable have a negative impact on Sharpe ratio. This negative effect is equally pronounced for both positive and negative variation in state variables. During this period Default Spread and Dividend Yield have very similar effect on Sharpe ratio.

Figure 4.6(c) shows the sensitivity plots for post-GFC period (2009-2013). Consistent with results in Figures 4.6(a) and 4.6(b), Trend and Dividend Yield have the largest impact on Sharpe ratio. However, there is a positive relation between these state variables and Sharpe ratio and their impact is large. For example, one standard deviation variation in these variables increases the Sharpe value by about 100 percent. Term Spread has a negative relation with Sharpe across the full range of variations in its value, while Default Spread has a positive relation with Sharpe for the extreme changes in its values.

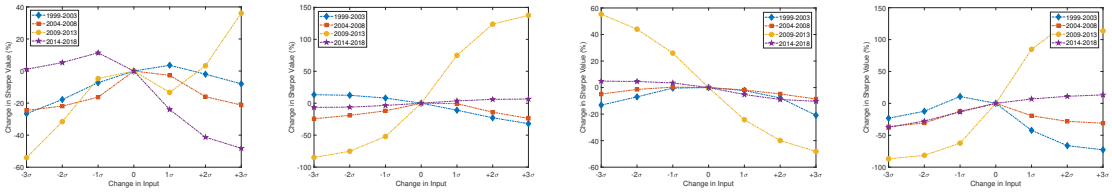
The sensitivity plots for the period 2014-2018 in Figure 4.6(d) is similar to those for the post-GFC period in Figure 4.6(c) where the relation between Trend and Dividend Yield, and Sharpe ratio is positive. However, their impact on Sharpe ratio is less, particularly for Dividend Yield for which Sharpe ratio varies less than 10 percent across its full range of variation. During this period, the largest Sharpe ratio range is for Default Spread, followed by Trend. Positive variation in Default Spread brings down the Sharpe ratio by close to 50 percent. While the extreme negative changes in Default Spread have less impact on Sharpe ratio. Term Spread has a negative relation with Sharpe ratio and its impact on Sharpe ratio is less compared to that of the post-GFC period.

In general, sensitivity plots in Figure 4.6 suggest that Trend (Term Spread) is the most (least) important state variable in predicting Sharpe ratio compared to other state variables. The direction of the impact of Term Spread on Sharpe ratio is different across sample periods. During the periods that include an economic turbulence any changes in Term Spread values decreases Sharpe ratio, while during the post-crisis

periods Term Spread has a negative relation with this ratio. Dividend Yield and Trend has similar impact on Sharpe ratio over these crisis, and their relations with Sharpe are positive during post-GFC and the recent period (2014-2018), however, the impact of Dividend Yield is not as strong as Trend for the most recent sample period (2014-2018). The changing range for Sharpe ratio is almost similar for both Default Spread and Dividend Yield for the subsample periods that include the economic crises. However, during the post-GFC period, Dividend Yield is the most important determinant of Sharpe ratio compared to Default Spread. Nevertheless, during the most recent period of 20014-2018, Default Spread is the most important state factor.

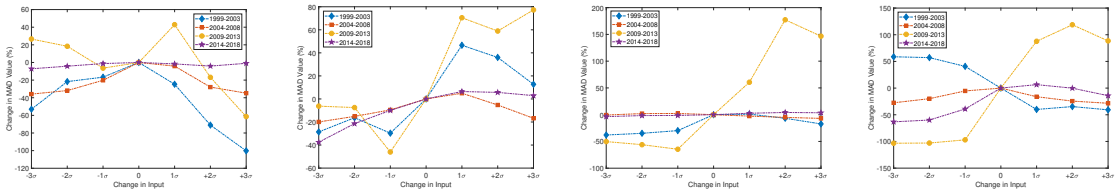
Our results from the sensitivity analysis is consistent with our finding in Table 4.4 where we found Trend as the most important determinant of Sharpe ratio, followed by Default Spread and Dividend Yield. Also, it provided further support for the finding that Default Spread is the most important variable for the recent period of 2014-2018.

4.9.3 Supplementary Results of Sensitivity Analysis



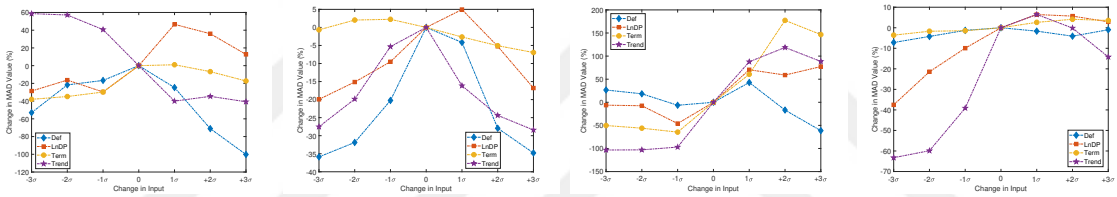
(a) Default Spread (b) Dividend Yield (c) Term Spread (d) Trend

Figure 4.7: Sensitivity analysis for Sharpe ratio



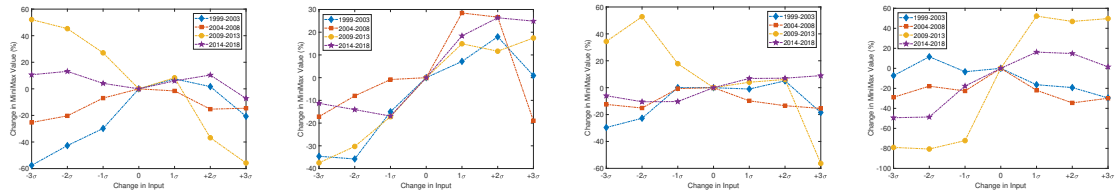
(a) Default Spread (b) Dividend Yield (c) Term Spread (d) Trend

Figure 4.8: Sensitivity analysis for MAD ratio



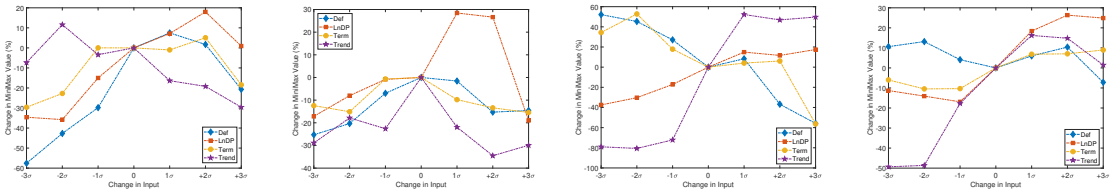
(a) 1999-2003 (b) 2004-2008 (c) 2009-2013 (d) 2014-2018

Figure 4.9: Sensitivity analysis for MAD ratio in different subsamples



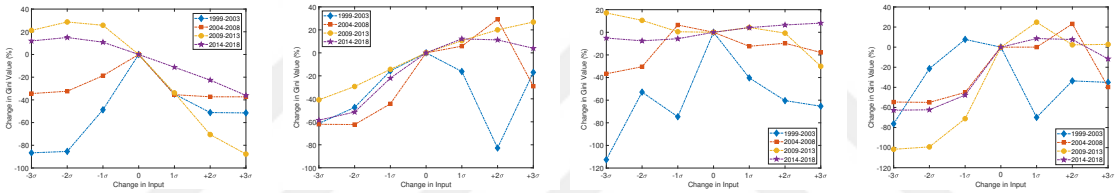
(a) Default Spread (b) Dividend Yield (c) Term Spread (d) Trend

Figure 4.10: Sensitivity analysis for MiniMax ratio



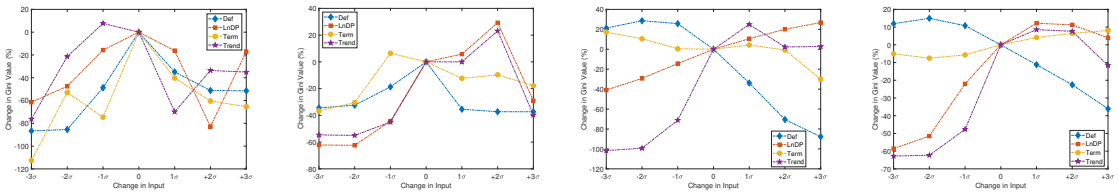
(a) 1999-2003 (b) 2004-2008 (c) 2009-2013 (d) 2014-2018

Figure 4.11: Sensitivity analysis for MiniMax ratio in different subsamples



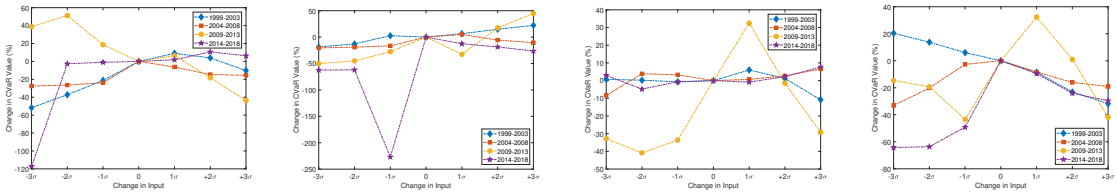
(a) Default Spread (b) Dividend Yield (c) Term Spread (d) Trend

Figure 4.12: Sensitivity analysis for Gini ratio



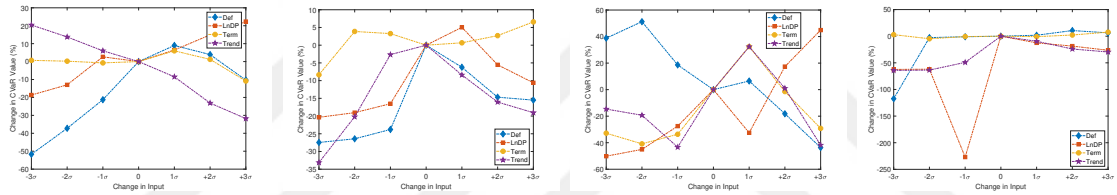
(a) 1999-2003 (b) 2004-2008 (c) 2009-2013 (d) 2014-2018

Figure 4.13: Sensitivity analysis for Gini ratio in different subsamples



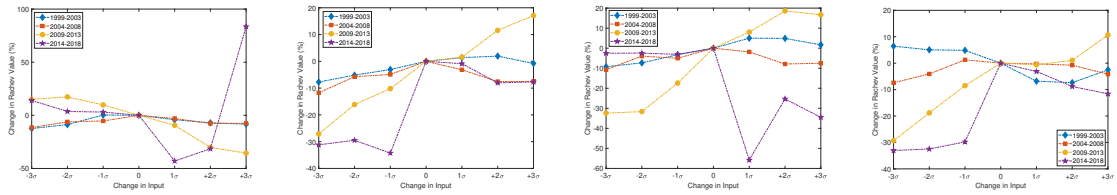
(a) Default Spread (b) Dividend Yield (c) Term Spread (d) Trend

Figure 4.14: Sensitivity analysis for CVaR ratio



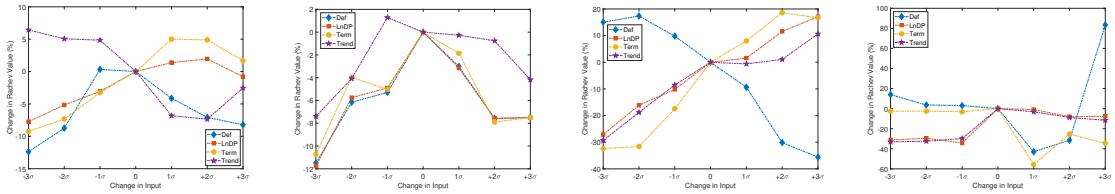
(a) 1999-2003 (b) 2004-2008 (c) 2009-2013 (d) 2014-2018

Figure 4.15: Sensitivity analysis for CVaR ratio in different subsamples



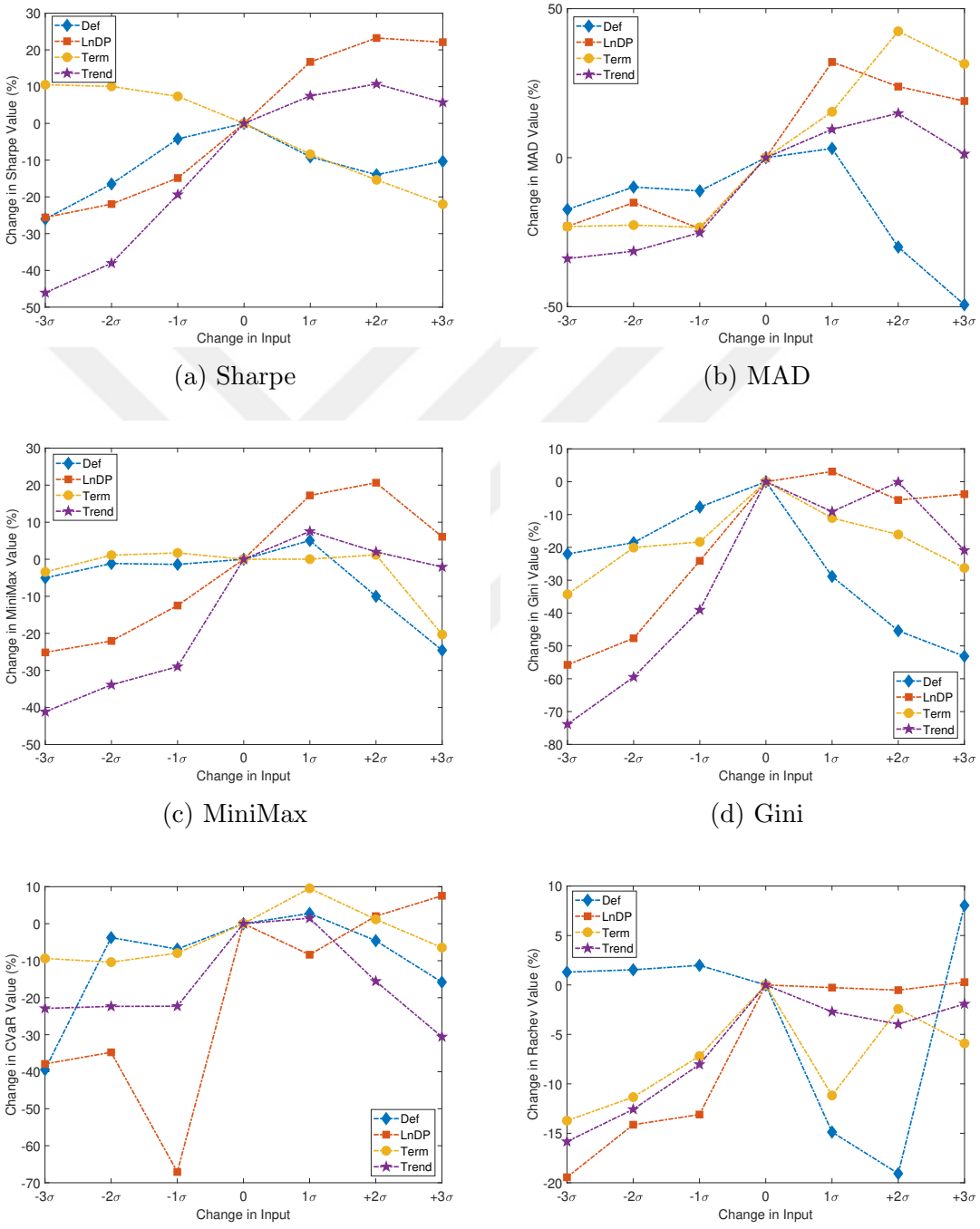
(a) Default Spread (b) Dividend Yield (c) Term Spread (d) Trend

Figure 4.16: Sensitivity analysis for Rachev ratio



(a) 1999-2003 (b) 2004-2008 (c) 2009-2013 (d) 2014-2018

Figure 4.17: Sensitivity analysis for Rachev ratio in different subsamples



(a) Sharpe (b) MAD

(c) MiniMax (d) Gini

(e) CVaR (f) Rachev

Figure 4.18: Sensitivity analysis for all ratios by change in state variables over the whole sample period 1999-2018

4.10 Time Varying Performance of ratios

It has been a long debate which performance ratio is the best to be used in the management process. We employ the optimal portfolio choice for each performance ratio predicted by DNN over the test sample period and shed some lights on this issue. Particularly, we investigate if there is a single performance ratio that outperforms other ratios for all periods by looking at the final wealth of associated optimal portfolios that we obtained using DNN.

For each ratio during the test period (240 months), we use the optimal weights based on the DNN method and compute monthly returns. Table 4.5 reports the average monthly returns for each ratio in panel (a) and the number of months that each ratio has generated the highest return (final wealth) compared to other ratios in panel (b). The results from panel (a) show that Rachev and MiniMax ratios generate higher monthly returns compared to other ratios, while the Sharpe ratio earns the lowest return. However, Rachev and MiniMax are not the absolute outperformers. Panel (b) shows that MiniMax and Rachev are the best ratios for 67 and 42 months out of 240 months, respectively, and other ratios perform better in other months. Interestingly, Sharpe is the best ratio for only 23 months and is ranked last of the six ratios. CVaR is only slightly more frequently successful than Sharpe with 27 months as the best ratio, but it ranks third for performance.

In Figure 4.19, we show which ratio outperforms others in each month. The success of MiniMax which has been the best ratio for 67 months are concentrated mostly during the periods of 1999-2008 and 2014-2018. This ratio has not been the best ratio during and after the Global Financial Crisis (GFC). On the other hand, CVaR and MAD perform the best during GFC, while Rachev and Gini outperform other ratios in post-GFC period.

These findings show that frequency and sequence of success for a performance ratio is not a good indicator of the outcome based on that ratio. In other words, their performance is time-varying.

To confirm this, we investigate the relationship of the frequency of a ratio with its

Table 4.5: Ranking of performance ratios. This table shows the ranking of performance ratios according to the average monthly returns and frequencies as the best ratio during the out-of-sample period. The table also reports the Pearson's correlation and its p-value (in parenthesis) between two rank series.

Rank	Panel (a): Monthly Accumulated Return (%)		Panel (b): Frequency	
	Ratio	Value	Ratio	Value
1	Rachev	1.08	MiniMax	67
2	MiniMax	1.02	Gini	42
3	CVaR	0.95	MAD	41
4	MAD	0.89	Rachev	40
5	Gini	0.85	CVaR	27
6	Sharpe	0.85	Sharpe	23
Correlation	0.31 (0.54)			N=240

Note. The correlation between two rank series is tested by Pearson's linear coefficient test, and the p-value reported in parenthesis shows the fail to reject null hypothesis that no correlation exists.

returns. We rank the ratios with respect to frequency as well as the monthly returns and report the Pearson's correlation rank in Table 4.5. The correlation is insignificant at conventional levels, suggesting that there is no relationship between these rankings. These results are different from the findings of Farinelli et al. (2008), who show that the leading ratios in terms of producing returns are the most frequent.

Our results in Table 4.5 suggest that the frequency of ratios is not a credible rule to guide the managers and they should not focus on a single performance ratio in their management process.

4.11 Dynamic Selection of Performance Ratios and Predicting the Optimal Weights

In this section, we propose an approach based on machine learning techniques to use the joint distribution of the market state variables and the realized historical performances of these ratios to accommodate the state-dependent preferences of investors.

Suppose we optimize several performance ratios using the proposed DNN approach

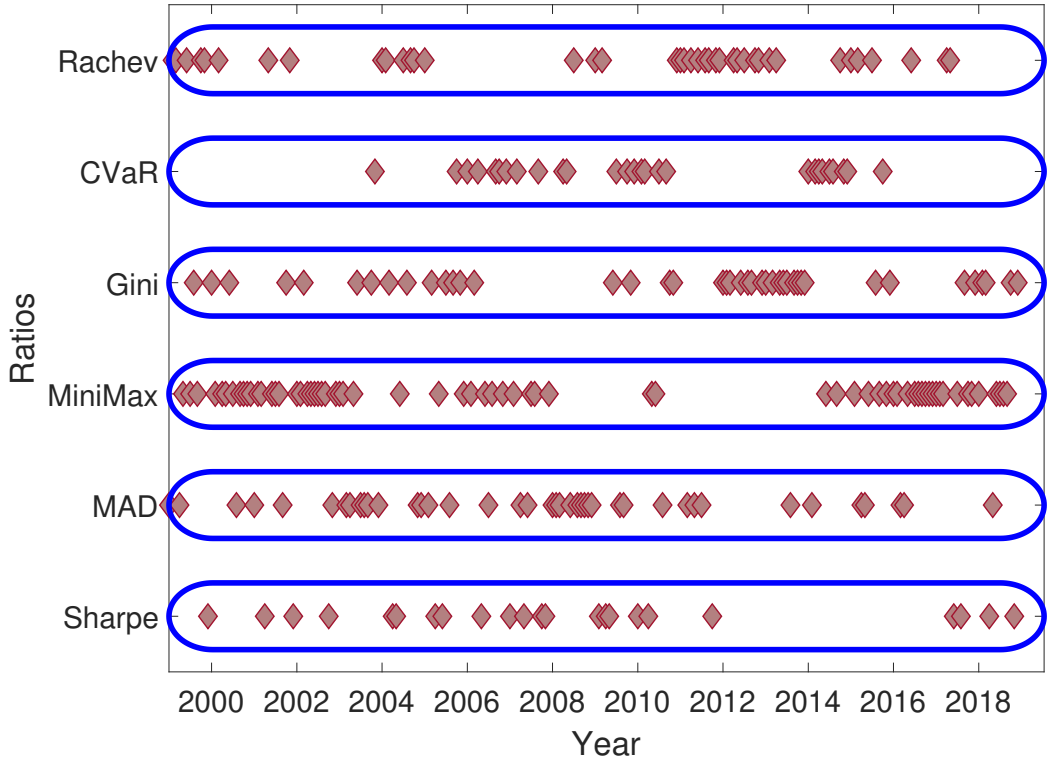


Figure 4.19: Sequence of the best ratio over time.

in Section 4.6, and observe which ratio offers a portfolio with higher return in each historical sample period. This observation provides us with pairs of state variables and the best ratio in each period. This pairing is known as 'labeling' in machine learning, and in this context, the best performance ratio is referred as the 'label'. We set up a DNN that learns the pattern between state variables and the best ratio labels during the historical sample period. This trained DNN allows us to choose the ratio that has the highest probability to outperform other ratios in a new period. Then, we use the trained predicted (DNN) model for this performance ratio to find the portfolio weights for the new period. We now explain the details of this dynamic selection of performance ratios by DNN.

4.11.1 Assignment of Market States to Ratios

In this study, we solve problem (4.18) for different performance ratios. For simplicity, let $u \in \{1, \ldots, K\}$ denote the set of performance ratios of interest, and $\hat{\Lambda^u}$ for $u = 1, \cdots, K$, represent the optimal DNNs (predicting models) after solving Problem (P1) for performance ratios $\hat{\psi^u}$ (as in Section 4.6). Given the state variables $\mathbf{z}_t$ as the network's input at the beginning of the historical sample period t, this set of trained DNNs gives different optimal portfolio weights for each performance ratio in each period, i.e. $\mathbf{x}_t \in \{\hat{\Lambda}(\mathbf{z_t})^{(u_t)} : u_t = 1, \cdots, K\}$. We use these sets of optimal weights and determine the best performance ratio (u_t^*) and its associated predicting model ($\hat{\Lambda}(\mathbf{z_t})^{(u_t^*)}$) that generates the highest return at the end of period t.

This will result in a time series of the pairs of state variable vector (as input) and the best performance ratio (label) over the historical sample period as:

$$\{(\mathbf{z}_1, u_1^*), \ldots, (\mathbf{z}_T, u_T^*)\} \tag{4.40}$$

Here, $\mathbf{z}_t \in \mathbb{R}^M$ and $u_t^* \in \{1, \ldots, K\}$ for $t = 1, \ldots, T$.

We design a DNN classifier, as a predicting model, to identify which performance ratio performs the best, conditioned on the state of the market at period t. DNN for classification is a learning model which uses sample observations of pairs (inputs and labels) for classification. DNN as a classifier can handle multiple classes, where data is classified into two or more for its labels. This is an important feature of DNN that makes it suitable for our purpose as we deal with multiple performance ratios as labels. In our DNN classifier, again the input is a vector of state variables, and the network has K output nodes which represent predicted probabilities for each label. Let define p_t^k as the actual label probability (or label indicator) with values values $p_t^k = 1$ if $u_t^* = k$, and $p_t^k = 0$ if $u_t^* \neq k$ for $k = 1, \cdots, K$. These probabilities are estimated by a softmax function in DNN classifier (Alpaydin, 2009; Heaton et al.,

2016) as in Equation (4.41)

$$\hat{p}_t^i = softmax(o_t^i) = \frac{\exp o_t^i}{\sum_k \exp o_t^k} \quad \forall i = 1, \cdots, K \tag{4.41}$$

where o_t^i is the output of the hidden layer in DNN and $\hat{p}_t^i$ is considered an approximate probability of label i given the state variable $\mathbf{z}_t$. A cross-entropy function is a suitable loss function to measure the performance of a classification-predicting network model since its output is a probability value between 0 and 1 (Li and Pi, 2019). Cross-entropy loss decreases as the predicted probability converges to 1, which indicates the actual label (u_t^*). We calculate the total loss of all labels over the historical periods using Equation (4.42):

$$l_{loss} = -\sum_t \sum_k p_k^t \log(\hat{p}_t^k) \tag{4.42}$$

where p_t^k is 1, if the actual label of observation$\mathbf{z}_t$ is k, and $\hat{p}_t^k$ is its predicted value. The DNN classifier aims to minimize loss function l_loss.

In the minimization algorithm, we use the gradient descent approach, which follows the positive direction of the gradients to minimize Function (4.42). A sequence of network weights based on random gradients are iteratively generated until there is no improvement in the function.

Once the network is trained, we have the predicting model that maps the market states in the new period to the performance ratio that has the highest probability to outperform other ratios in that period. Therefore, to select the most suitable performance ratio for a new period with the given state variable we use the trained classifier.

4.11.2 Predicting Optimal Weights by the Best Performance Ratio

To determine optimal weights ($\mathbf{x}_{T+1}$) in the new period $T+1$, we use the trained DNN of the best ratio for that period and the state variables to find the optimal portfolio

weights. More formally:

$$\mathbf{x}_{T+1} = \hat{\Lambda}(\mathbf{z}_{T+1})^{(u^*_{T+1})} \tag{4.43}$$

4.12 Suggested Procedure of Optimal Portfolio Prediction

In summary, in our proposed procedure to find optimal weights without estimations of returns moments, we conduct the following steps. First, we train a DNN network for each performance ratio, given the state of the market, and find its associated model to predict portfolio weights. Second, we train a DNN to find a classifier model that predicts which ratio outperforms the others in the new period. Third, we enter the state variables into the DNN classifier, identify the best ratio, and then use its associated DNN to find the optimal portfolio weights for the new period. Figure 4.20 shows a schematic diagram for this procedure.

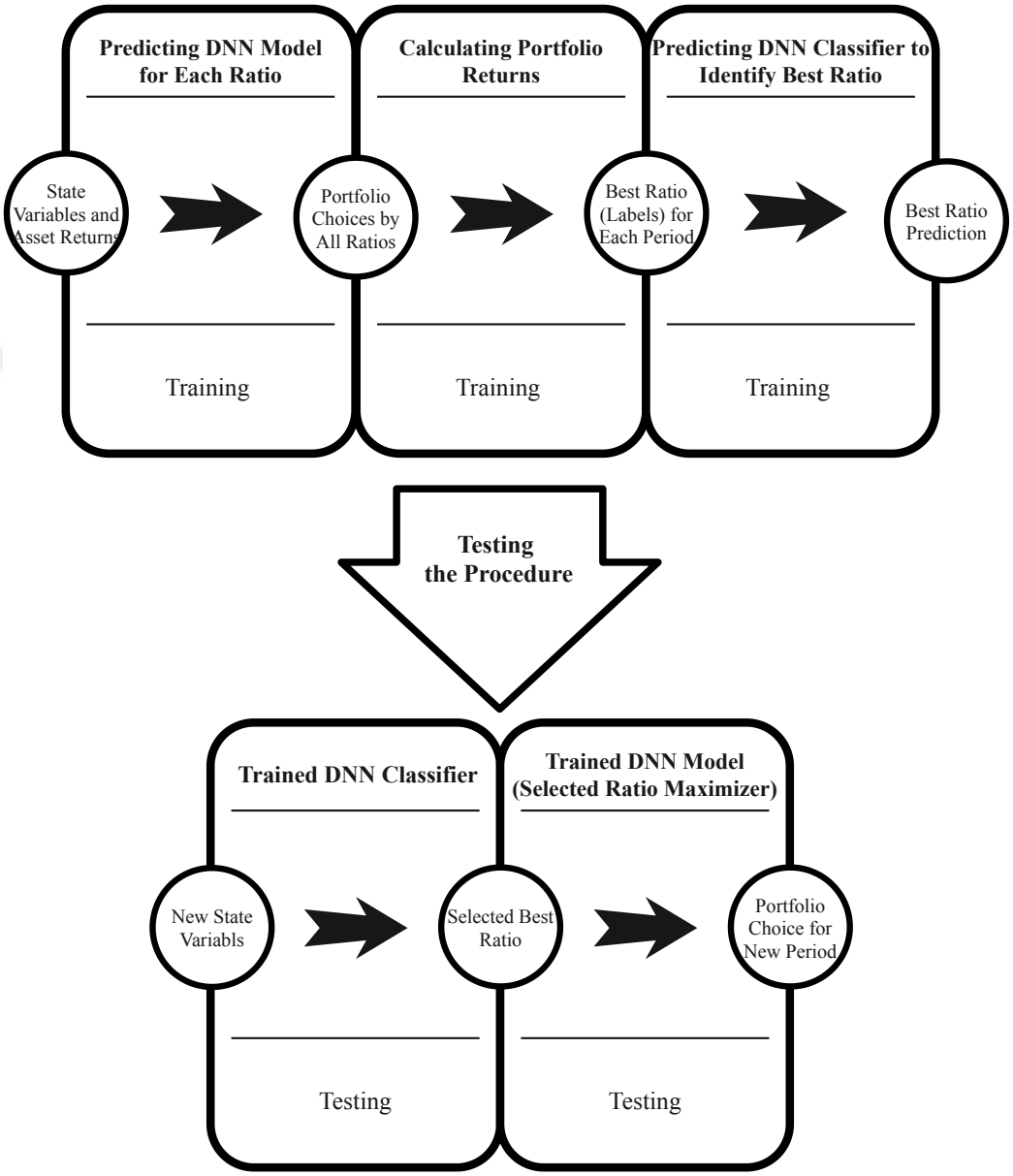


Figure 4.20: The schematic diagram of the proposed procedure for portfolio choice

4.12.1 Dynamic Portfolio Rebalancing

In the second step of the proposed procedure, we select the best ratio for each month of the training sets. Then, we use the combination of the best ratio and the state of the market to train a DNN classifier, as a predictor, that uses the state variables at the beginning of the month and predicts which performance ratio is most likely to

outperform other ratios in that month.

In the third step, having developed the two DNN predicting models using the training sets, we employ the DNN classifier and state variables each month over the 240-month test period. We identify the ratio that has the highest chance to be the leading ratio based on the state of the market each month. Then we use the maximization DNN associated with that ratio and the state variables to find the optimal weights of our portfolio for each month over the test period. Table 4.6 reports the monthly average returns over the test period. Column 1 in Table 4.6 presents the returns using our proposed method of selecting ratios dynamically during the test period. Columns 2 to 7 in Table 4.6 show the results when we use a single ratio to optimize the portfolios.

The average monthly return using our proposed method is 1.47%, which is higher than those based on single ratios. The closest one is Rachev, which generates 1.08% on monthly basis, 39 basis points lower than our method. Interestingly, the Sharpe ratio, a commonly used performance measure, performs worst among the single ratios and our dynamic selection outperforms Sharpe by about 72 basis points.

We also analyze the performance of our method during important market episodes and present the results for different sub-samples of the test period. These periods are associated with important economic and financial turbulence. The 1999–2003 period includes the Dotcom bubble, 2004–2008 started with a bull market and ended with the Global Financial Crisis (GFC), and 2009–2013 is associated with the post-GFC period. Finally, 2014–2018 includes the recent growth period. Dynamic selection of ratios performs better than selecting any single ratio in all sub-samples. This suggests that our proposed method is robust in different episodes of the market.

Interestingly, the sub-sample results for single ratios show that using Sharpe or MAD does not lead to the highest return in any of the sub-samples, while asymmetrical ratios generate higher returns. This is because they take into account heavy tails in the return distribution. This makes it possible to hedge investment against heavy losses while taking advantage of positive shocks in returns (Farinelli et al., 2008).

Table 4.6: Dynamic selection of performance ratios versus a single ratio in estimation of portfolio choice.

Period	Monthly Accumulated Return (%)						
	DR (1)	Sharpe (2)	MAD (3)	MiniMax (4)	Gini (5)	CVaR (6)	Rachev (7)
All sample	1.4725	0.8514	0.8927	1.0203	0.8519	0.9512	*1.0842*
	(3.4163)	(3.5572)	(3.5786)	(3.3324)	(3.3499)	(3.2532)	(3.3610)
1999-2003	1.3608	0.5603	0.7046	0.9220	0.8048	*0.9503*	0.8856
	(3.2273)	(2.9845)	(3.4799)	(3.0276)	(2.9133)	(3.1215)	(3.0654)
2004-2008	1.1951	1.1633	1.0237	0.9114	*1.1812*	0.8167	0.8179
	(3.5168)	(3.7339)	(3.7476)	(3.8047)	(3.5706)	(2.8146)	(2.8181)
2009-2013	2.1668	1.3396	1.2573	1.2061	1.1768	1.5448	*2.0734*
	(4.1597)	(4.4180)	(4.0986)	(3.9332)	(4.2053)	(4.0827)	(4.4977)
2014-2018	1.1674	0.3423	0.5853	*1.0418*	0.2446	0.4932	0.5600
	(2.5526)	(2.8631)	(2.9391)	(2.4114)	(2.4377)	(2.8189)	(2.6088)

However, none of the single ratios lead across all sub-samples. CVaR, Gini, Rachev, and MiniMax produce higher returns for the sub-sample periods of 1999–2003, 2004–2008, 2009–2013, and 2014–2018, respectively. CVaR and Gini generate the highest returns during 1999–2003 and 2004–2008, respectively. Rachev outperforms the others during the post-GFC period, which includes the European sovereign debt crisis. This provides further support to the literature that shows that in the presence of low/negative returns, and positive shocks, Rachev captures large drawdowns and positive shocks by a suitable risk measure and reward function and generates higher gains compared to other ratios (Rachev et al., 2007; Adcock et al., 2019).

Standard deviations of the returns for the dynamic selection method, as well as for single ratios, are higher during the first and second sub-samples than the others, as both of these periods include crises that lead to widely fluctuating returns. In summary, the dynamic switching of ratios guides the investor to choose the best ratio conditioned on the state of the market, which in turn delivers the optimal portfolio weights based on the selected ratio and improves the portfolio return remarkably.

Figure 4.21 shows the cumulative growth of $1 invested in portfolios with different

methods for the selection of performance ratios used in the DNN optimization problem. The cumulative growth is computed for the whole test period (January 1999 to December 2018). We consider seven identical portfolios which are rebalanced by different ratios maximized by DNN as well as the suggested dynamic selection method by DNN. Plot paths show one dollar investment growth which is cumulatively calculated through different rebalancing strategies over the whole sample test period. The suggested dynamic method gets dominant from the 17th month onward and never drops below any of the six paths generated by the well-known ratios. The dynamic selection method generates the highest wealth compared to methods in which different single ratios have been used. The end of the period wealth for the dynamic method ($29.20) is more than double that of Rachev as the closest measure ($11.66).

Figure 4.21 shows that the relative performance of the portfolios is time-varying. However, the portfolio that is rebalanced using the dynamic method outperforms other portfolios very early in the sample and continues to outperform until the end. Other portfolios do not show such a consistent performance. This suggests that the proposed method eliminates the effect of time on the return distribution and efficiently takes the greatest advantage of the true distribution of asset returns. Our suggested method dominates the method in which one single ratio is used, and hence, the associated portfolio consistently outperforms other portfolios.

So far, we have used DNN to optimize the ratios when we compare a single ratio method with our proposed dynamic approach. The reason for this is that, as Table 4.2 shows, DNN performs better than traditional methods in maximization of ratios. However, as an illustration, we also compute the monthly returns of portfolios optimized using traditional methods as well as our proposed procedure, which includes a dynamic selection of performance ratios and uses DNN for the optimization. As discussed in Section 2, in traditional methods, some models such as VAR or factor models are used to estimate the moments of returns. These moments, then, are used in a single ratio for optimization.

For each ratio, we use the best traditional method, either AR (1) or the factor

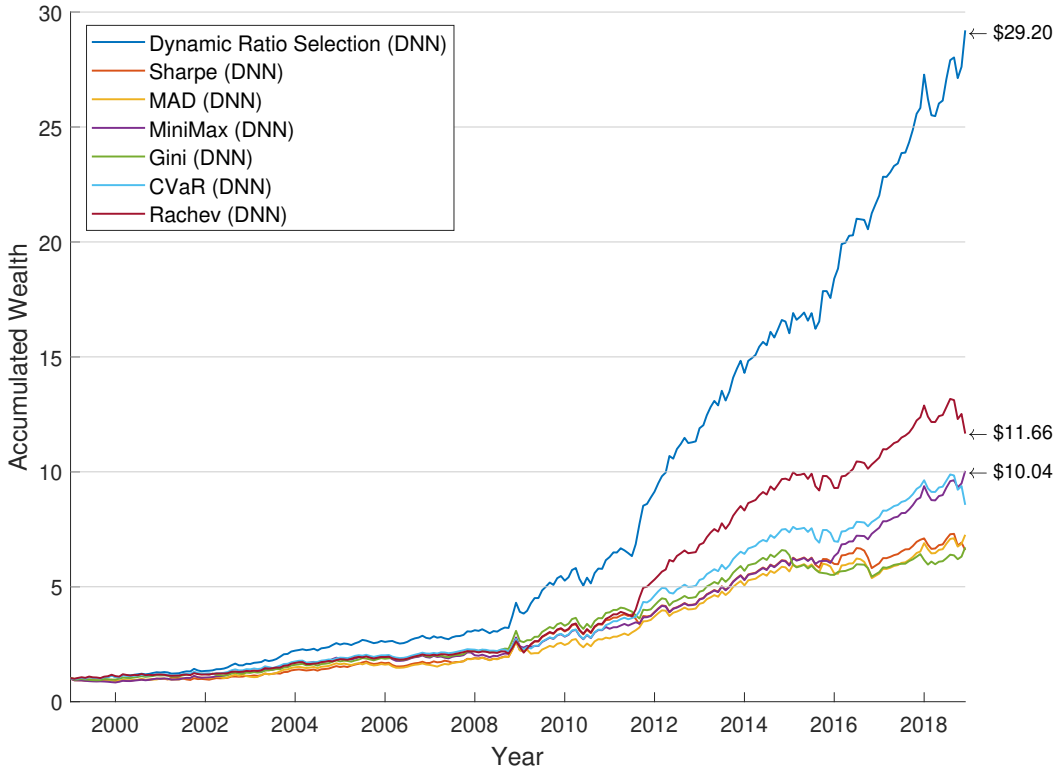


Figure 4.21: Growth of $1 investment at the beginning of the test period using each ratio maximization by DNN as well as the suggested dynamic selection method.

model, that yields higher a value for the ratio and optimizes our portfolio. Table 4.7 reports the monthly average returns. The proposed procedure generates a return of 1.47% per month on average over our entire test sample and outperforms traditional approaches using various ratios. The mean difference between our proposed method and the traditional techniques varies 0.6–1.33% per month on average. We further compare the cumulative growth of $1 invested in portfolios using our proposed procedure and traditional methods in Figure 4.22. We consider seven identical portfolios which are rebalanced by different ratios maximized by the benchmark method as well as the suggested dynamic selection method by DNN. Plot paths show one dollar investment growth which is cumulatively calculated through different rebalancing strategies over the whole sample test period. The suggested dynamic method gets dominant from the 17th month onward and never drops below any of the six paths generated by the well-known ratios. The end of the period wealth for the dynamic method ($29.20) is about four times higher than that of Gini ($7.05) and Sharpe ($6.41) as the closest ratios.

The proposed procedure allows investors to take the greatest advantage of extreme positive returns and hedge against huge negative downturns by using the ratio that is most likely to beat other ratios. Also, using DNN instead of traditional methods in this approach captures nonlinear relationships between state variables, returns, and risk.

Table 4.7: Optimal portfolio returns: dynamic selection of performance ratios versus traditional estimates of portfolio choice.

Method	DR	Sharpe	MAD	MiniMax	Gini	CVaR	Rachev
Mean	1.47^{***}	0.82^{***}	0.72^{***}	0.15	0.87^{***}	0.70^{***}	0.33
	(6.68)	(4.55)	(2.84)	(0.59)	(4.00)	(2.83)	(1.37)
Mean Difference		0.66^{***}	0.75^{***}	1.33^{***}	0.60^{***}	0.77^{***}	1.14^{***}
		(3.55)	(4.60)	(5.24)	(2.88)	(3.29)	(4.38)

Note. Significance levels refer to the t-test for mean difference between DNN and the best of benchmarks. $^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

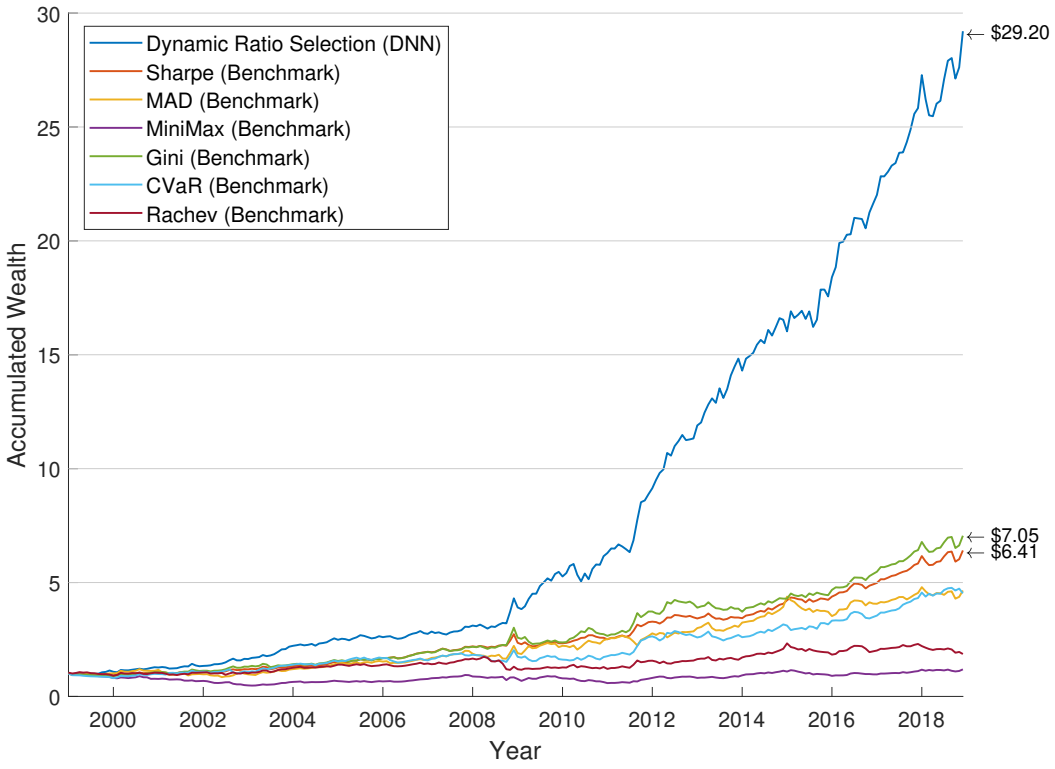


Figure 4.22: Growth of $1 investment at the beginning of the test period using each ratio maximization by benchmark method as well as the suggested dynamic selection method.

In this study, we have used four state variables as the predictor of the return distribution. It is interesting to examine which state variable explains the returns in our proposed approach. We regress the monthly realized returns of portfolios optimized using our dynamic selection approach method on the lagged state variables. The state variables are demeaned and standardized. The limitation in this analysis is that we cannot interpret the slopes of this linear regression as factor loading because the DNN has captured the nonlinear relationships. However, this analysis indicates the relative role of these state variables (Colombo et al., 2017).

Table 4.8 shows the regression results. The results show that the Default factor explains the variations in returns of the portfolio using the dynamic selection approach. We also report the results when single ratios are used in the optimization. The coefficient of the Default factor is significant for all ratios.

Table 4.8: State variables and optimal portfolio returns.

State Variable	DR	Sharpe	MAD	MiniMax	Gini	CVaR	Rachev
Default Spread	1.24***	0.89**	0.99***	0.80**	0.79**	1.15***	1.40***
	(3.61)	(2.44)	(2.69)	(2.33)	(2.28)	(3.45)	(4.15)
Dividend Yield	0.08	0.22	0.08	0.09	0.12	-0.14	0.04
	(0.32)	(0.81)	(0.24)	(0.33)	(0.45)	(-0.56)	(0.14)
Term Spread	-0.108	-0.04	-0.16	0.04	-0.17	-0.03	-0.16
	(-0.43)	(-0.16)	(-0.59)	(0.15)	(-0.67)	(-0.12)	(-0.63)
Trend	0.36	0.41	0.60*	0.28	0.32	0.80***	0.92***
	(1.16)	(1.27)	(1.82)	(0.92)	(1.04)	(2.67)	(3.03)

Note. Significance levels refer to the t-test for mean difference between DNN and the best of benchmarks. $^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

The positive and significant coefficient of Default spread as a systematic risk factor is consistent with the notion that extreme returns occur along with higher values for the risk factor. The Trend is another factor that predicts returns for all ratios except Sharpe, MiniMax, and Gini. Both Trend and Default are important factors when CVaR, MAD, or Rachev, as a single ratio, is used in the optimization problem.

4.13 Conclusion

The main contributions and findings of this chapter are as follows:

- A novel approach to conditional asset allocation is suggested using DNN, which builds a complex nonlinear model to generate optimal portfolio weights directly using state variables.
- The robustness of the proposed DNN-based method is presented using empirical data and various performance ratios.
- Interpreting the outcome of the estimated models by analytical methods shows that investors should not rely on a single or a specific combination of state variables in their portfolio advice.
- Time-varying portfolio performances of different ratios suggests that investors should not rely on a single measure for portfolio optimization.

Chapter 5

FIRM EVALUATION IN THE U.S. MARKET

In this chapter, we focus on the pricing efficiency in initial public offering (IPO). IPO is the process of introducing a private firm to a market for the first time. Evaluation of these firms is a challenging task due to the lack of historical information on trading and subsequently insufficient knowledge of their past performance. We propose a nonlinear approach based on stochastic frontier and neural networks to estimate the pricing efficiency and the level of premarket inefficiencies.

5.1 Literature Review

In the following, we provide a review of the pertinent literature. This is presented in three parts. First, we review the literature on underpricing theory. Then, we review the related work on the methods including new statistical approaches have been introduced to address this problem, and finally, review the assiciated factors in pricing and underpricing.

5.1.1 Underpricing Theory

It is well documented that Initial Public Offerings (IPOs) are, on average, underpriced on the first day of trading Aggarwal et al. (2009); Carter and Manaster (1990); Miller and Reilly (1987); Ibbotson et al. (1988); Ritter (1987); Smith Jr (1986). Underpricing in IPOs is ascribed to the asymmetry of information surrounding the value of these shares between the different participants in the market, including the issuer, underwriter, and outside investors. Reber and Vencappa (2016) state that asymmetrical information leads to classic adverse selection problem (Akerlof, 1978) in new issues, which reveals itself through persistent average initial return across capital market.

The literature suggests various explanations for the underpricing, which can be classified under two broad hypotheses. First, premarket deliberate underpricing, which refers to the outcome of information asymmetry among participants who are involved in premarket evaluation. These information are mainly classified into two types of firm attributes and deal characteristics. Baron (1982) presents a theory which clarifies that the issuers are not aware of efforts and marketing of underwriters. Thus, underwriters compensate their reputation and expertise in signaling the quality of the issuing firm by the underpricing mechanism. The model of Rock (1986) supposes that underpricing is to induce less-informed investors to participate in the new issues market. Welch (1989) and Allen and Faulhaber (1989) describe the underpricing as a tool for high quality firms to signal to outside investors, while it is costly for low quality firms to imitate such signaling. The second type of explanation accounts for the underpricing of trading activities in the aftermarket. Aggarwal and Rivoli (1990) investigate the impact of fads on overvaluation of IPOs in early aftermarket. The effect of demand for a new share is another approach which explain the initial return, (Ritter, 1991; Friesen and Swift, 2009; Chahine, 2007; Koop and Li, 2001).

Second, some researchers attribute the initial return to aftermarket mispricing (Arba and Karaa, 2006; Chen et al., 2002). However, some others, for example, Francis and Hasan (2001), Hunt-McCool et al. (1996), and Grinblatt and Hwang (1989) still support the dominant belief that the contribution of deliberate underpricing is more than aftermarket mispricing. Hence, the literature attributes a part of an initial return to deliberate underpricing. A strand of the literature investigates this phenomenon in both premarket and aftermarket to disentangle the underpricing into deliberate and aftermarket mispricing (Hunt-McCool et al., 1996; Reber and Vencappa, 2016). Therefore, first day initial return is built up of two parts, that is, a portion made by asymmetry information in premarket and the other portion is made by aftermarket activities once the trading on an equity starts, along with the revelation of information to all participants. This part of the initial return could be positive or negative depending on the aftermarket conditions. However, Hunt-McCool

et al. (1996); Reber and Vencappa (2016) report a very low correlation between first day initial return and estimated deliberate underpricing meaning that deliberate underpricing is not necessarily the main component for initial return in IPOs. These conflicting results make this issue remain still an open question in IPO valuation.

5.1.2 Analysis Methods

The separation of deliberate underpricing and aftermarket mispricing, two possible components in explaining the initial return, is complicated without having a systematic upper bound for the fair price which is not observable. Accordingly, one could identify each component by having an estimate of the fair price, in addition to the offer price and closing price in the aftermarket. Thus, it is assumed that the offer price is less than or equal to the fair price. This is the main logic behind the method of Stochastic Frontier Analysis (SFA), employed by Hunt-McCool et al. (1996) for IPOs. In this parametric method, the difference between the fair price and the offer price is made up of two error terms. A one-sided normal error term which captures the deliberate underpricing, plus a second normal error which completes the nonsystematic deviations from the fair price. Reber and Vencappa (2016) disentangle the level of deliberate underpricing and aftermarket mispricing by investigating the effect of proxy variables on each component. Hence, they develop SFA by relating the variance of the deliberate underpricing distribution to variables which systematically determine the level of deliberate underpricing. They suppose that the fair price as a frontier is achievable by relying on some pricing factors which explain the IPO price performance. Pricing factors include proxy variables such as EBITDA and the book value of assets which influence the performance, following also, for instance, Peng and Wang (2007) and Aggarwal et al. (2009). In this chapter we build the assumptions about proxy variables and their relationships with dependent variables based on the study of Hunt-McCool et al. (1996) and Reber and Vencappa (2016).

Classical methods of statistics are based on low order of proxies to the number of observations. In these methods, inferring rules use some assumptions such as

Central Limit Theorem which supposes that the estimated statistics from a sufficient large sample will be approximately close to those of the population. Accordingly, some predefined and well known distributions are needed in order to formulate a model. Different distributions have one or more parameters to be estimated from an observation sample. In the problem of IPO and its literature the models used so far are mainly based on classic methods. Ordinary Least Square (OLS), Generalized Linear Models (GML), and non-linear models (logit) are such methods. SFA is one of the successful methods applied in IPO valuation which is also a parametric model estimated by Maximum Likelihood (ML). The distributional assumptions in SFA as a parametric method, is a drawback. For example, half-normal, exponential, gamma and truncated normal distributions are used for systematic error term of deliberate underpricing in SFA. So far all the relations between components and proxies have been considered as linear in SFA or at most are transformed by log function. In spite of many reports on linearity among variables in the literature, we are interested in broader types of dependencies in this research. The relationships and patterns between variables might be more complicated than what a simple OLS and linear regression or even restricted nonlinear models could find out. Moreover, complex cross sectional issues among data make the economical results and interpretations be conflicting in different studies using linear methods. In spite of simplicity of traditional methods and interpretability of their results, some disadvantages like the limitation of these methods in modeling and capturing the biased observations or outliers lead to emergence of modern statistical methods. Use of new statistical methods and machine learning techniques is not new in Finance. Several studies apply new models in IPO as a real word problem in this area. Robertson et al. (1998); Reber et al. (2005); Esfahanipour et al. (2016) develop Neural Network (NN) to predict first-day initial returns of IPOs. They compare this method with traditional models like regressions and show the explanatory power of (NN) in prediction. Jain and Nag (1995) apply neural network model to price IPOs and present significant economic gains using this method. Luque et al. (2009) develop a two-layered evolutionary system to predict

IPO underpricing. They compare the suggested method against several alternatives and the results indicate the relative strong performance of the proposed method.

5.1.3 Factors

Several proxy variables associated with each IPO are intended which are investigated in previous studies in sake of exploring their economic relations with offer price as the dependent variable of IPOs. In what follows as an explanation for proxies, we mainly focus on the most studied variables considered in recent IPO researches.

Pricing Factors

A firm is fundamentally valuated by estimating its future expected cash flows, cost of capital, and growth opportunities (Reber and Vencappa, 2016). This estimation would be doubly hard task for an analyst valuating an IPO. This is due to the lack of enough information about financial performance of a firm equity going public for the first time. To overcome this issue, several proxy variables have been introduced as drivers of an equity value of a firm based on standard finance theory. We define and review the selected important variables in this context and restate the results of the studies which examine the IPO pricing problem with these variables.

The first pricing factor is EBITDA (Earning Before Interest, Tax, Depreciation and Amortization). Earnings are a good reflector for future cash flows and in long term earnings converge to cash flows (Teoh et al., 1998; Reber and Vencappa, 2016). However, some studies use EPS for IPO valuation [1], recent research consider EBITDA that is more reliable than EPS which is not a relevant variable in IPO pricing (Koop and Li, 2001). The other pricing factor which presents future growth of a firm is negative earnings before flotation of the stock. Aggarwal et al. (2009) investigate the impact of negative earnings on firm valuations and find that firms with more negative (positive) earnings have higher valuations than those firs with less negative (positive)

[1] Chen et al. (2002); Koop and Li (2001); Luo and Ouyang (2014); Peng and Wang (2007) are such research.

earnings. Following Reber and Vencappa (2016) who find significant relationship between negative earnings and IPO value, we use this as a dummy variable which is equal to 1 for IPOs acquiring the most recent earnings negatively and 0 otherwise. One of the important segments and activities of a firm is Research and Development (R&D) section. Corporations innovate to develop new technologies or improve their current services and products. According to this, R&D expenses suggest for future growth of a firm which have undertaken progression plans toward more advanced state. Reber and Vencappa (2016); Aggarwal et al. (2009); Hertzel et al. (2012); Deeds et al. (1997) incorporate it into IPO valuation. In order to capture the differences such as risk and investment opportunities across various industries, Hunt-McCool et al. (1996) use 6 industry sectors. We follow Koop and Li (2001); Reber and Vencappa (2016) to consider 12 industry sectors based on two-digit classification defined by Fama and French (1997) as dummy variable representing a specific business type to which an IPO goes public. Reber and Vencappa (2016) use the book value as a lower bound for IPO value. Hunt-McCool et al. (1996); Koop and Li (2001); Chen et al. (2002); Peng and Wang (2007) use book value as a factor in valuation. Finally, lon term debt or leverage that indicates the level of risk and chance of bankruptcy is as a proxy in valuations (Reber and Vencappa, 2016). Search corroborate positive relationship between EBITDA, R&D spending, book value, and firm value, while it is negative for negative earnings and leverage.

Underpricing Factors

We include underpricing factors supposed by Reber and Vencappa (2016) representing information asymmetry between different parties engaged through different stages of IPO valuation process. Two proxies for underpricing are sales and age of the firm. Hunt-McCool et al. (1996); Koop and Li (2001); Peng and Wang (2007); Aggarwal et al. (2009); Reber and Vencappa (2016) find positive association between sales and firm value. Hunt-McCool et al. (1996); Chen et al. (2002); Reber and Vencappa (2016) use age as a proxy for risk capturing ex ante uncertainty. First study finds a positive

association between age and IPO value while two others findings suggest a negative correlation. It is assumed that a firm with lower sales and age have higher premarket risk and makes the issuers to underprice in order to compensate less informed investor for an adverse selection problem (Reber and Vencappa, 2016). In IPO process, a commitment guarantees a proportion of shares not to be sold by stockholders immediately after going public. Equity retention is a costly signal to transmit firm value from insiders to outside investors, since it reduces diversification of owners' personal portfolio (Reber and Vencappa, 2016). Hunt-McCool et al. (1996); Roosenboom and van der Goot (2005) find a positive relationship between equity retained and IPO value, while Aggarwal et al. (2009) report it negatively. Reber and Vencappa (2016); Chen et al. (2002) do not find a statistical significant relations, that's why it is omitted in some research (Koop and Li, 2001; Luo and Ouyang, 2014). Proceeds at the disposal of the IPO is another underpricing factor measured by the number of initial shares times the offer price per share. This is regarded as another signal for potential investment plans and hence a higher value. Hunt-McCool et al. (1996) report a positive association between proceeds and offer price, and Reber and Vencappa (2016); Peng and Wang (2007) find that proceeds is negatively related to underpricing. Koop and Li (2001); Chen et al. (2002); Aggarwal et al. (2009) exclude proceeds from their estimations. Lock-in agreement is a commitment that prohibits insider from selling shares in specified period (90 days to two years) after going public. We use a dummy variable equal to 1 if an IPO has lock-in and 0 otherwise. The impact of this variable is on the share price as in the presence of this agreement the demand for a new issue is more that offers compared to IPO without this commitment (Reber and Vencappa, 2016). Brav and Gompers (2003); Arthurs et al. (2009); Reber and Vencappa (2016) report a positive relations between lock-in dummy variable and pricing inefficiency. Underwriting fees is another proxy for risk of an IPO as the underwriter compensation is due to information cost and deal characteristics (Hughes, 1986). Therefore, fees are higher for less known and younger firms. Regarding correlation between fees and offer price, Hunt-McCool et al. (1996); Reber and Vencappa (2016) find it a positive relationships

while Koop and Li (2001) report negative correlation. some studies do not incorporate fees in estimations (Chen et al., 2002; Aggarwal et al., 2009). Number of proceeds disclosed at flotation prospectus is considered as the other deal feature which indicates pricing inefficiency due to information asymmetry. Beatty and Ritter (1986); Reber and Vencappa (2016) report a positive correlation between proceeds number and ex ante uncertainty while, in contrast, Ljungqvist and Wilhelm Jr (2003); Leone et al. (2007) find a negative relationship. Regarding underwriter, its reputation is a proxy for underpricing. Baron (1982) supposes that underwriters have more comprehensive information than issuers then IPOs are prone to be underpriced by underwriters to compensate their reputation which could signal outside investors about the value of the firm. We follow intuition of Reber and Vencappa (2016) for this variable who use underwriter rankings in SFA to explain deliberate underpricing for the first time. They do not find statistically significant relationships between underwriter ranking and pricing inefficiency. We consider a dummy variable for IPOs with Private Equity and Venture Capital (PE/VC) backing. Reber and Vencappa (2016) test the impact of this proxy on deliberate underpricing in SFA context for the first time. Several research show that PE/VC backing describes firm value (Arthurs et al., 2009; Barry et al., 1990; Megginson and Weiss, 1991; Bartling and Park, 2009; Brav and Gompers, 1997; Krishnan et al., 2011; Lerner, 1994; Nahata, 2008; Nanda and Rhodes-Kropf, 2013). Some mixed results exist about IPO backed by EP/VC like Gompers (1996) states that IPOs with younger EP/VC backing have higher initial returns while Liu and Ritter (2011) report an opposing result that is IPOs covered by more reputable analysts have higher initial return. We use a dummy variable indicating the market condition of new issues following the definition considered by Yung et al. (2008); Reber and Vencappa (2016). Accordingly, the market condition at the time of flotation is classified into Hot, Cold, and Normal. They use aggregate number of IPOs and aggregate initial returns to recognize the market state ,since previous research show that these two quantities are positively correlated (Lowry, 2003; Loughran and Ritter, 2004; Brailsford et al., 2004; Yung et al., 2008). In contrast to previous studies which

report cyclicity in both number of IPOs and initial returns, Reber and Vencappa (2016) conclude that deliberate underpricing is independent of IPO market condition. The last underpricing factor is also a dummy variable that studies introduce it as an effort toward capturing the IPO cycles (Lowry and Schwert, 2002; Loughran and Ritter, 2004; Ritter, 1984). Amount of change in real private nonresidential fixed investment is proxy for firms' demand for capital. Reber and Vencappa (2016) define three periods similar to market condition for capital demand based on this quantity. It is assumed in the liturature that some shocks in the market leads to increase the activities in new issue market which result in low quality IPOs and more information asymmetry leading to higher initial returns in Hot periods Reber and Vencappa (2016). Yung et al. (2008); Reber and Vencappa (2016) find that the level of price inefficiency is lower in Hot periods of capital demand. They state that this evidence is strange and more investigation is required.

In this study, we follow the instruction of Yung et al. (2008) to determine Hot, Cold, and Normal condition with respect to the number of IPOs, underpricing, and demand for private capital. We consider three variables of the number of IPOs, average initial returns, and percent change in Private Nonresidential Fixed Investment (PNFI) all on quarterly basis. For each variable we compute the moving average MA(4) in each quarter along with the historic average in all previous quarters going back to 1975. If this moving average is 50% above (below) the historical average, the quarter is classified as hot (cold). All the remaining quarters are considered as normal.

Atermarket Mispricing Factors

A part of the literature devotes the initial return to aftermarket activities (Arba and Karaa, 2006; Chen et al., 2002). Accordingly, some proxies which affect the demand for share directly and indirectly have been investigated. The most used variables include: 1) The number of traded shares on the first day. A positive association is reported between this proxy and initial return Chahine (2007). 2) Likelihood of flip-

ping defined by two dummy variables which are equal to one if the offer price is above or below the initial filing price range. This variable captures aftermarket demand in IPO by the price revision Reber and Vencappa (2016). Ellis (2006) shows that there is statistically significant relationships between these two dummy variables representing being above and below the filing price range and trading volume positively and negatively, respectively. 3) The portion of firm stock held by insiders as a signal to outside investors is another proxy for which a negative dependency with initial return is reported Reber and Vencappa (2016). 4) Underwriter reputation as another variable affects on demand in two ways, first, the IPOs going public by more reputable underwriter are more interesting for investors and second, these IPOs take advantage of price support by the underwriter in aftermarket. Reber and Vencappa (2016) find a significant and positive relationships between initial return and reputation and Ellis (2006) reports a positive association between reputation and trading volume. 5) Offer size is considered as a risk of the firm that a smaller IPO is riskier than those which are larger and more established. Reber and Vencappa (2016) find that offer size is positively related to initial return while others like Beatty and Ritter (1986); Carter (1992) report a negative association.

5.2 Method

In the next section, we design a DNN based on the SFA approach. Thus, we first briefly present the SFA model in IPO pricing and then construct a DNN using the problem specification similar to SFA.

There is a well known and empirically successful approach for estimation of inefficiency by the method of Stochastic Frontier Analysis in IPO pricing. SFA is an econometric method which was simultaneously proposed first by Aigner et al. (1977) and Meeusen and van Den Broeck (1977) in stochastic production frontier. Their work is related to the estimation of productive efficiency by frontier production functions. This approach aims to explain the gap between a predetermined input level and output efficiency. The stochastic frontier model with single-equation production

function is written as (Battese and Coelli, 1995; Luo and Ouyang, 2014):

$$\ln Y_i = a + X_i'\alpha + v_i - u_i, \quad i = 1, \cdots, N \tag{5.1}$$

Where Y_i is the offer price for the i-th firm at the time of floating. X_i is the vector of K explanatory variables of the i-th firm. v_is are assumed to be independent and identically distributed variables and represent the usual statistical noise. u_is are non-negative random variables, associated with the technical inefficiency of the firms. There are two general approaches to estimate the inefficiency by SFA. First, Ordinary Least Square (OLS) approach may be applied to (5.1) (Schmidt and Sickles, 1984). An extension to OLS approach named Corrected Ordinary Least Square (COLS) is introduced by Farrell (1957) and Lee and Schmidt (1993) which involves shifting the line toward the best performing company. The COLS procedure shifts the frontier up by the amount at which the minimum inefficiency would be zero or equivalently there is at least one firm with %100 efficiency. The generated frontier envelopes the whole data. Thus the one sided error term estimations is obtained as follows:

$$-u_i^* = \hat{u}_i - \max(\hat{u}_i) \tag{5.2}$$

and the COLS intercept is estimated as:

$$a_{\text{COLS}} = a + \max(\hat{u}_i) \tag{5.3}$$

The second approach is based on independence of errors from regressors, and specific distributional assumptions have been made for v and u which are usually normal for v and half-normal for u. Several studies have extended the model by applying other well known distributions for u such as exponential (Meeusen and van Den Broeck, 1977), truncated normal (Stevenson, 1980), gamma (Greene, 1990), Weibull (Tsionas, 2007), genaralized gamma (Griffin and Steel, 2008), and combination of distributions (Griffin and Steel, 2008). Maximum likelihood estimation is derived and described

for this approach by Pitt and Lee (1981) and used by Hunt-McCool et al. (1996) in IPO pricing for the first time.

We follow the former approach, which has the major advantage of letting inefficiencies be distribution free as a desirable characteristic and it allows for correlation between inefficiencies and firm specific explanatory variables. However, this robustness is obtained at the cost of losing the underlying identity of u_i (Greene, 2005), we assume a considerable value on capturing highly nonlinear relationships between variables that this method provides us. In addition to the variables X_i in equation (5.1), we consider a set of variables which explain the inefficiency following Reber and Vencappa (2016). We reformulate the equation (5.1) as follows which is proposed by Neghab et al. (2021a):

$$\ln Y_i = f(X_i; \mathbf{w_1}) - g(Z_i; \mathbf{w_2}) + v_i, \quad i = 1, \cdots, N \tag{5.4}$$

where, $f(X_i; \mathbf{w_1})$ and $u_i = g(Z_i; \mathbf{w_2})$ are no longer linear functions in this study. We estimate unknown parameters, $\mathbf{w_1}$ and $\mathbf{w_2}$ within complicated functions that captures highly nonlinear relationships between response and explanatory variables. Given the estimates of $\mathbf{w_1}$ and $\mathbf{w_2}$, we can recover efficiency in pricing for i-th firm defined by the following equation (Battese and Coelli, 1995):

$$E_i = \exp(-u_i^*) \tag{5.5}$$

where u_i^* is estimated by equation (5.2). Once adjustment is made for premarket inefficiencies, $1 - E_i$, and fair offer price, $P_i^* = Y_i/E_i$, are recovered, the remaining part of the initial first day return r_i as aftermarket mispricing (Reber and Vencappa (2016) define it as the difference between market price and fair offer price) is regressed on some other aftermarket explanatory variables S_i. Then, we have:

$$P_i - P_i^* = q(S_i; \mathbf{w_3}) + \epsilon_i \tag{5.6}$$

Here, P_i is close price or market price, P_i^* is the estimated fair offer price, and function $q(S_i; \mathbf{w_3})$ is unknown too and the parameters $\mathbf{w_3}$ have to be estimated.

Hunt-McCool et al. (1996) were the first to apply the maximum likelihood method to the problem of IPO pricing. Further, they assume that there is an efficient frontier for the price of an IPO. Thus, the difference between actual offer price and the maximum price is modeled by SFA. This gap as an economic inefficiency is explained by the error terms mentioned in the model of Aigner et al. (1977). Their findings show the dependency between underpricing and the market state, and conclude that higher abnormal returns and premarket underpricing exist in hot market periods. A relatively weak relationship between measured underpricing and excess returns prevent them explaining abnormal returns. They then suggest that some other factors, including underwriter reputation, consumer confidence, or market power on the part of underwriter, are required to complete the puzzle. Consequently, they state that premarket underpricing and aftermarket mispricing may exist simultaneously. Koop and Li (2001) examine the pricing of IPOs and seasoned equity offering (SEO) firms by SFA. They also model the misvaluation distribution in the pricing equation as a function of observable firm- and issue period-specific properties. They found that the variables included to explain misvaluation are mostly insignificant. However, the dummy variable used in labeling an SEO or an IPO is reported as a highly significant variable, indicating that IPOs are more underpriced relative to SEOs. Chen et al. (2002) investigate the IPO pricing in the presence of noisy trading activities in Taiwan. They choose the market trading and firm characteristics as variables to explain noisy trading and examine the relationship between noisy trading, and IPO price performance. Their results show that the initial returns are highly affected by noisy trading activities and it is more significant in a hot market than in a cold market. Peng and Wang (2007) employ SFA in the pricing of IPOs in Taiwan. They present a lower underpricing for IPOs characterized by greater earning potentials, less risk or less asymmetric information. Moreover, a statistically significant relation is reported using their selected proxies of underpricing. The average values of underpricing for

in-sample, and a group of IPOs which use the auction method, imply that the introduction of the auction method decreases underpricing in IPOs. Luo and Ouyang (2014) use Bayesian Stochastic Frontier Analysis (BSFA) to study IPO pricing efficiency in China. Their empirical findings report an average of IPO pricing inefficiency equal to 0.5908 and conclude that this low amount might be attributed to inefficient work of underwriters. Finally, they categorize the effective variables into two sets which are positively and negatively related to pricing inefficiency. The recent work of Reber and Vencappa (2016) uses SFA to recognize the level of deliberate underpricing and aftermarket mispricing in initial returns. They utilize the related factors for each dependent variable which are the fair price, deliberate underpricing, and aftermarket mispricing. In their model, they simultaneously estimate the parameters of fair price and deliberate underpricing by SFA. The average deliberate underpricing is reported at 14% along with an average of 19.1% for the observed initial returns. They then find that a portion of the initial return is explainable by deliberate underpricing using a separate regression. Once they estimate the deliberate underpricing and subtracted from the initial return, they regress the remainder as aftermarket mispricing on its associated factors. Furthermore, their analysis reveals that the fraction of deliberate underpricing is more than that of aftermarket mispricing in making up the initial return.

In the next section, we introduce a machine learning tool and then modify it to be applicable to this problem.

5.2.1 Designing a DNN for the IPO Problem

In the IPO problem we have a set of independent factors for each component of an IPO which could be considered as input variables. The response variables are $\ln Y_i$ and $P_i - P_i^*$ which are considered as output variables. Figure 5.1 shows a network with three subnetworks where each one transmits the information to obtain the desired output. Subnetwork 1 is fed with pricing factors and gives an estimate of the fair offer price f. Subnetwork 2 is fed with underpricing factors and gives an estimate

of function g which is equivalent term of deliberate underpricing. By subtracting the final value obtained in subnetworks 2 from that of subnetwork 1, an estimate of $\ln Y_i$ will be obtained in an output unit based on equation (5.4). After adjustments for efficiency and computing aftermarket mispricing component $P_i - P_i^*$, a separate network is fed with aftermarket factors and gives an estimate of aftermarket mispricing q. We represent the network parameters by $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$. These are the weights associated with the network edges for subnetworks 1, 2 and 3, respectively. In hidden layers, we may have various numbers of units. We use sigmoid function in hidden nodes as activation function.

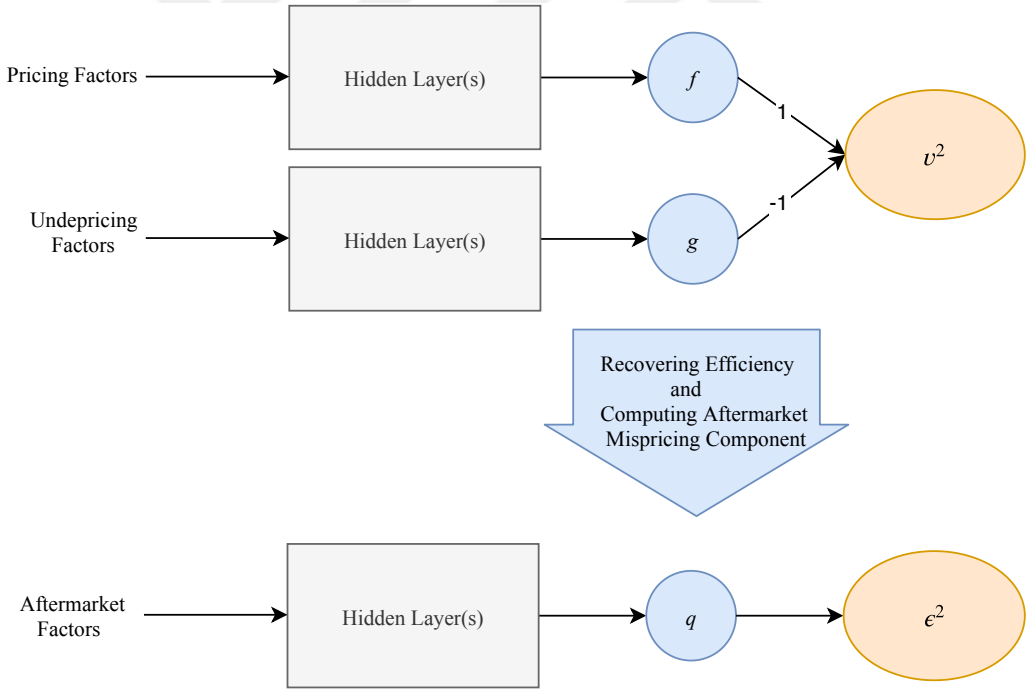


Figure 5.1: DNN for IPO

The Feed-Forward and Backpropagation Algorithm

The calculations of this algorithm will be presented in the context of the IPO problem as follows.

Let consider the combination of subnetworks 1 and 2. In this network, inputs are pricing and underpricing factors, X_i and Z_i, respectively. The output is an estimation for $ln(Y_i)$ which is the log of the offer price for the i^{th} IPO and calculated by adding two output estimated values of f and $-g$ through the network. The goal is to minimize the error term v in equation (5.4). The objective function for all the error terms is the expectation of all errors of all N IPOs:

$$\mathbb{E}\left[v^2|X,Z\right] = \frac{1}{N}\sum_{i=1}^{N} v_i^2 \tag{5.7}$$

Given the proxies for sample i, we can write v^2 as a function of parameters with the weights of connections in the DNN as $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2\}$:

$$v_{i\,|X_i,Z_i}^2 = \mathcal{Q}_i(\mathbf{w}) \tag{5.8}$$

Finally, the objective function will be:

$$\mathcal{Q}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{Q}_i(\mathbf{w}). \tag{5.9}$$

Finding the zeros of the function (5.9) by the Backpropagation algorithm (BP) is of interest. This algorithm in DNN is based on stochastic approximation methods. These methods are common when the critical point of a complex objective function cannot be computed directly, but might be estimated by noisy observations. Stochastic problems are optimized by the iterative method of Stochastic gradient descent (SGD). The algorithm starts with an initial random value of $\mathbf{w}$. The following equation performs the iterations converging to the optimal point by providing the next solution:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \gamma\Delta\mathcal{Q}(\mathbf{w}_n) \tag{5.10}$$

Here γ is the step size, or learning rate, of the algorithm, and $\Delta\mathcal{Q}(\mathbf{w}_n)$ denotes the partial derivative of the function evaluated at $\mathbf{w}_n$ as n^{th} solution.

The Feed-forward step in the DNN, which is explained in Section 5.2.1, is equivalent to evaluating the objective function at a random initial value of $\mathbf{w}$. Once this step is complete for all samples, the BP algorithm begins. In this step, the aim is to compute the partial derivative of the function $\mathcal{Q}$ with respect to each element of $\mathbf{w}$. As the Feed-forward is a recursive substitution of computed functions at DNN nodes followed by the transfer of information through linear transformation of the nodes' outputs, the BP is implemented by the chain rule for computing derivatives. This means that we should follow all paths to a parameter backward and calculate the derivatives to reach that parameter on an edge in the DNN.

Consider Figure 5.1. This Network has $L = 3$ layers including one input layer, hidden layer, and output layer, from left to right. These are numbered from 1 to 3. We need to compute derivatives of the objective function with respect to the weights connecting layers. Let $w_{ij}^{(l-1,l)}$ denote the weight between the i^{th} node of layer $l-1$ to the $j^{(th)}$ node of layer l. Let $net_i^{(l)}$ and $o_i^{(l)}$ denote the input and output of the $i^{(th)}$ node in layer l, respectively.

It should be noted that to pass a node, we need to compute the derivative of its output with respect to its input. Thus, for a node using a sigmoid as a computing function we have:

$$\frac{\partial o_i^{(1)}}{\partial net_i^{(1)}} = o_i^{(1)}(1 - o_i^{(1)}) \tag{5.11}$$

Here $o_i^{(1)} = s(net_i^{(1)})$, where s is a sigmoid function. Now to calculate the partial derivative $\frac{\partial \mathcal{Q}}{\partial w_{ij}^{(l-1,l)}}$, first we need to compute the derivative of $\mathcal{Q}$ with respect to outputs. Then:

$$\frac{\partial \mathcal{Q}}{\partial f} = -2(\ln(Y) - \ln(\hat{Y})) \tag{5.12}$$

$$\frac{\partial \mathcal{Q}}{\partial g} = 2(\ln(Y) - \ln(\hat{Y})) \tag{5.13}$$

To simplify the notation define a backpropagation error for node i in layer l and

denote it by $\delta_i^{(l)}$. Thus, for two output nodes we define:

$$\delta_1^{(3)} = \frac{\partial \mathcal{Q}}{\partial \ln(\hat{Y})} \frac{\partial \ln(\hat{Y})}{f} = -2v \tag{5.14}$$

$$\delta_1^{(3)} = \frac{\partial \mathcal{Q}}{\partial \ln(\hat{Y})} \frac{\partial \ln(\hat{Y})}{g} = 2v \tag{5.15}$$

As the subnetworks are separate from this layer backwards, we use each backpropagation error $\delta_k^{(3)}$ to compute the remaining derivatives with respect to the weights connecting nodes in subnetwork k, where $k = 1, 2$. Let $w_{ij}^{(l-1,l)_k}$ denote the weight between node i of layer $l-1$ to node j of layer l in subnetwork k. Then:

$$\delta_i^{(l-1)_k} = \sum_{j=1}^{n_k^l} w_{ij}^{(l-1,l)_k} \delta_j^{(l)_k}.$$

Here n_k^l is the number of nodes in the l^{th} hidden layer of subnetwork k. Finally, the required partial derivative is:

$$\Delta w_{ij}^{(l-1,l)_k} = \frac{\partial \mathcal{Q}}{\partial w_{ij}^{(l-1,l)_k}} = \delta_j^{(l)_k} o_i^{(l-1)}.$$

With an appropriate step size and the computed partial derivative, the following equation gives:

$$w_{ij}^{(l-1,l)_k} = w_{ij}^{(l-1,l)_k} - \gamma \Delta w_{ij}^{(l-1,l)_k}.$$

Once all the weights are updated, the next iteration starts by feeding the samples through the updated network.

The procedure is the same to train subnetwork 3 in which the inputs are aftermarket mispricing factors S, output is aftermarket component q and the objective function is to minimize the mean squared of error term for all IPOs, i.e. $\frac{1}{N} \sum_{i=1}^{N} \epsilon_i^2$.

5.3 Simulated Experiment

In this section, a set of synthetic data is generated. We test the model on simulated data, where the actual deliberate underpricing level is known and defined as the ratio of deliberate underpricing and the fair price. Arbitrarily, we choose two factors for each of pricing, deliberate underpricing, and aftermarket mispricing denoting by Vy_i, Vu_i, and Va_i, respectively, for $i = 1, 2$. We set the factors as follows:$Vy_1 \sim \mathcal{N}\ (\mu = 30, \sigma = 8)$, $Vy_2 \sim \mathcal{U}\ [L = 30, U = 60]$, $Vu_1 \sim \exp\ (\mu = 5)$, $Vu_2 \sim \mathcal{U}\ [L = 5, U = 10]$, $Va_1 \sim \mathcal{N}\ (\mu = 5, \sigma = 5)$, and $Va_2 \sim \mathcal{U}\ [L = -1, U = 5]$. Here $\mathcal{N}$, $\mathcal{U}$, and exp denote the normal, uniform, and exponential distributions, respectively. The total number of samples is equal to $N = 200$. The dependent observations are computed as follows: $y = 3Vy_1 + 0.5Vy_2 + \epsilon_y$, $u = 0.1Vu_1^2 + 2\sqrt{Vu_2} + \epsilon_u$, $a = 1.5Va_1 + Va_2 + \epsilon_a$, $op = y - u$, and $R = u + a$.

Here $\epsilon_y \sim \mathcal{N}(0, 2)$, $\epsilon_u \sim \mathcal{N}(0, 3)$, and $\epsilon_a \sim \mathcal{N}(0, 0.5)$ are normal error terms. All parameters and coefficients are selected in a way which gives reasonable magnitudes for observations, for example, positive values for prices. A notable point when generating u is that we assume that it is truncated at zero. Thus we set nonpositive values to zero. Following the definition, let $L_u = \frac{u}{y}$ denote the level of deliberate underpricing. Then we denote the estimated level by $\hat{L}_u = \frac{\hat{u}}{\hat{y}}$. Figure 5.2 shows the frequency of offer price and deliberate underpricing. We select 150 samples for training and the remaining for tests. The algorithm stops after the maximum number of iterations of 1000 with step size $\gamma = 0.0001$.

Figure 5.3 presents the mean of the squared errors in total for $\mathcal{E}$ by MSE_{Total}, and the mean of squared errors for the level of deliberate underpricing, defined as $MSE_{L_u} = (L_u - \hat{L}_u)$. We observe that as DNN moves towards the optimal point of $\mathcal{E}$ it also minimizes the errors in the level of deliberate underpricing.

Finally, Figure 5.4 indicates the prediction for the level of deliberate underpricing. Figure 5.4 (b) depicts the scatter of actual and predicted values of L_u, along with the reference line of 45°. It is observed that the model renders an acceptable level of deliberate underpricing for each IPO only with one assumption, that is, the deliberate

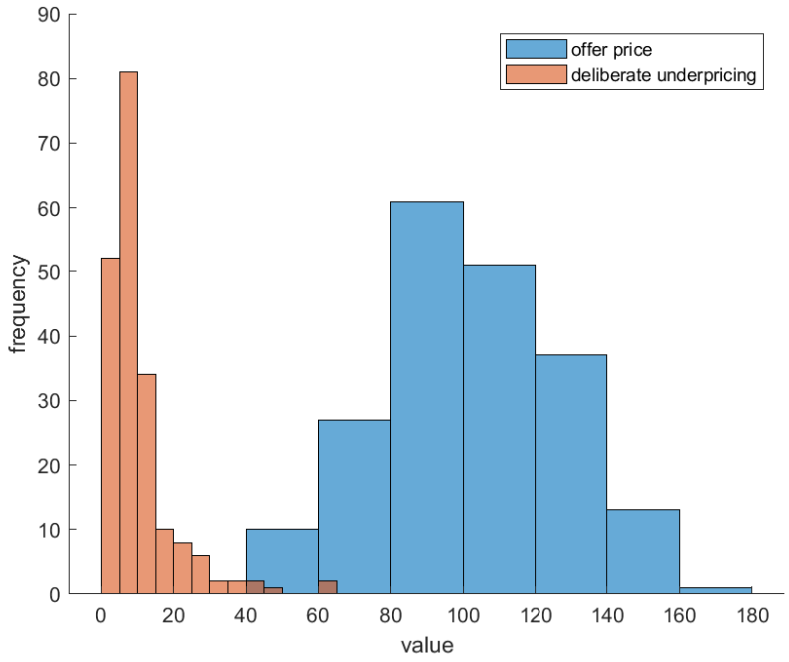


Figure 5.2: Frequency of offer price and deliberate underpricing

underpricing level is a one-sided distributed free variable truncated at zero. Results could be improved by implementing the algorithm with different parameters of the network, including step size, number of iterations, and random initial weights.

5.4 Data

We collect the information of U.S. firms going public in the new issues market. We obtain 1343 IPOs from October 1979 to November 2018 sourced with the financial transaction information from the Securities Data Company (SDC) Platinum. The financial statement data are obtained from the Compustat, and the trading data of each IPO from the Centre for Research in Security Prices (CRSP). We use Jay Ritters's website in order to find the age of firms at the time of listing, rankings on underwriter reputation, and the number of IPOs along with their average first day returns. We obtain data on private non-residual fixed investment which represent

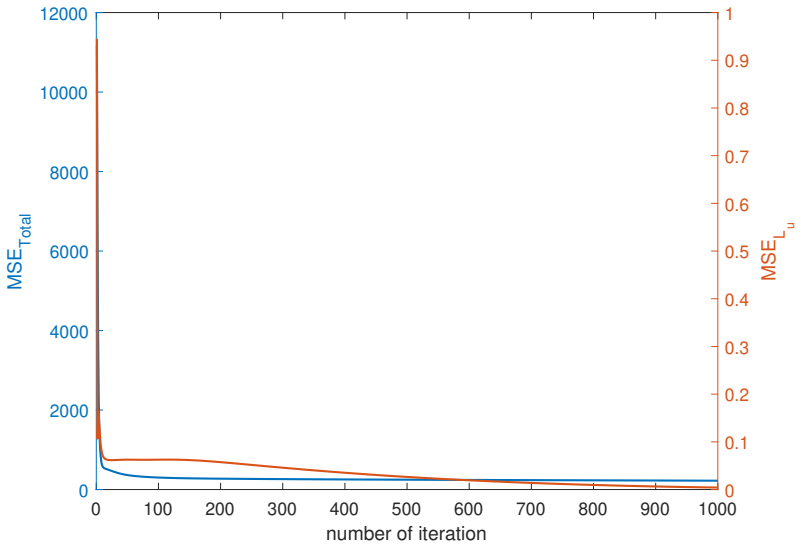
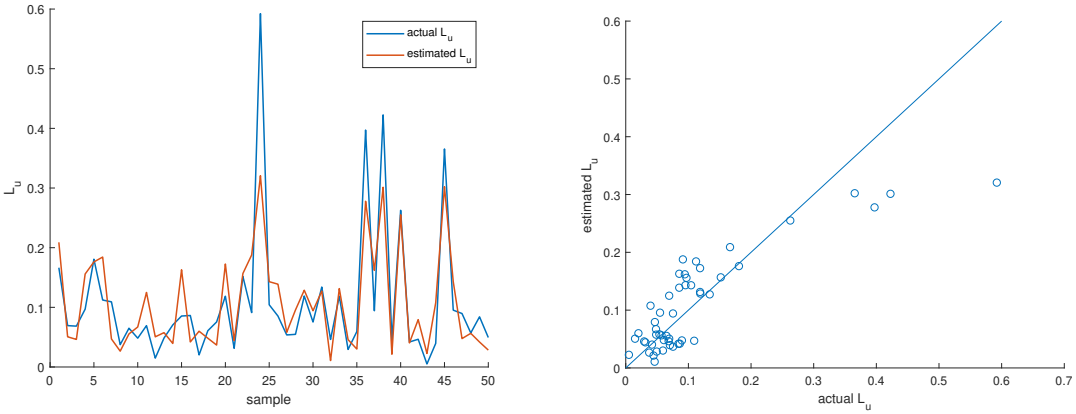


Figure 5.3: Mean Square Error (MSE). Total MSE (right axis) and deliberate underpricing level MSE (left axis)

firms' demand for capital from the Federal Reserve Bank of St. Louis. We set aside any IPO with missing values of any variable after merging all above data bases.

Table 5.1 presents the summary statistics for 1343 IPOs. Dependent variables



(a) Actual along with predicted (b) Scatter plot of actual against predicted

Figure 5.4: Out of sample prediction of L_u

are offer price and initial return which occur on the first day of trading. Offer price has a close mean and median with a small skewness and a kurtosis near 3 which all suggest a normal distribution. Average initial return is remarkably positive, however our data include 141 IPOs with negative initial return. In this table, independent variables include pricing factors of the firms which imply potential growth in future and possible cash flows they may produce accordingly. A positive average of EBITDA shows that firms are profitable before going public, however there are 338 IPOs with negative EBITDA. Positive skewness of 17.3 confirms that some of the firms have huge earnings at the time of flotation. Average of the dummy variable NI shows that 34.2% of the firms have negative income by listing date. Firms spend $3.7 million in R&D section on average. This variable is positively skewed while has a large kurtosis equal to 132.1. About 300 IPOs have not spent for capital expenditure. The average of assets is $80 million and it has a positive skewness with a large standard deviation equal to $405 million. Half of the IPOs have a quite low ratio of long term debt to asset which is 0.08. There is only one firm with leverage of 1, while 234 firms have no debt in accounting period before listing.

IPOs in our sample data, on average, are characterized by an age of 15.3 years. Age distribution is positively skewed, and a median of 8 years which is less than the average causes a relatively large kurtosis of 10. Sales as the other firm feature shows a large deviation and kurtosis. Although the average is $107.5 million, half of the firms have a sale less than $24.5 million. A dummy variable is assigned to separate industries. It is based on alphabetic order where the variable starts with 1 for "Chemical product" industry and ends with 12 for "Utilities" and value 13 stands for "ALL others". Regarding the variables which feature the deal at the time of flotation, owners retain more than half of the shares after listing. The negative value of -1.9 for skewness confirms that the owners tend to keep most of their shares after going public. Stockholders of only 22 firms have sold all of their shares. Proceed amount is $34.7 million on average. In about 54% of IPOs, shareholders make a commitment not to sell any shares over a specified period after going public. In

Table 5.1: Summary statistics

	Mean	Median	StdDev	Skewness	Kurtosis
Dependent variables					
Offer price ($)	11.872	12	4.474	0.611	3.593
Initial return (%)	15.534	7.500	25.466	3.750	26.005
Independent variables					
Pricing factors					
EBITDA ($million)	10.846	2.916	48.446	17.296	425.941
NI	0.342	0	0.474	0.667	1.445
R&D ($million)	3.661	1.406	9.464	9.702	132.125
Assets ($million)	80.758	17.515	405.599	21.512	576.601
Leverage	0.190	0.086	0.229	1.368	4.189
Sales ($million)	107.466	24.552	352.387	11.0190	187.928
Industry dummy	3.165	2.893	2.769	0.232	1.516
Underwriter fees (%)	1.615	1.545	0.399	3.797	26.066
Uderpricing factors					
(Firm characteristics)					
Firm age	15.296	8	19.545	2.624	9.930
(Deal characteristics)					
Equity retained*	0.660	0.691	0.162	−1.973	8.189
Proceeds ($million)	34.668	21.600	57.531	8.714	119.788
Lock in	0.538	1	0.499	−0.151	1.023
VC/PE backing	0.485	0	0.500	0.058	1.003
Underwriter reputation*	7.329	8.001	1.943	−1.494	4.556
Hot Market	0.864	1	0.342	−2.130	5.536
Cold Market	0.001	0	0.039	25.855	669.502
Hot underpricing	0.034	0	0.182	5.122	27.231
Cold underpricing	0.131	0	0.338	2.187	5.781
Hot demand for capital	0.021	0	0.143	6.707	45.986
Cold demand for capital	0.290	0	0.454	0.923	1.853
Aftermarket mispricing factors					
Offer size ($million)	43.494	26.400	71.084	6.855	71.351
Shares traded (%)	67.275	58.552	44.229	1.251	7.624
Above filing range	0.248	0	0.432	1.167	2.363
Below filing range	0.228	0	0.420	1.298	2.684

* This variable is also as of aftermarket mispricing factor.

our sample, 48% of the firms have venture capital and/or private equity backing. We report underwriter fees spent relative to offer price. Fees average is 1.6%. The average

of underwriters reputation is 7.32. The firms in our sample are mainly underwritten by reputable brokerages as we observe a median larger than the average value and a negative skewness for reputation. A considerable number of IPOs in our sample are issued during hot market activity. However, the condition with respect to level of underpricing and demand for capital is normal. Aftermarket activities for IPOs are briefed in four variables. Offer price with a positively skewed distribution and an average of $43.5 million. Percent of shares traded on the first day is equal to 67% on average, and the majority of IPOs offer price are in the initial filing range.

Table 5.2 presents the number of IPOs in each business sector in our sample data with respect to SIC code. Most of the IPOs are the firms with activities related to "Computers". In the section of "Oil and Gas", there is the fewest number of IPOs.

Table 5.2: Distribution of the number of IPOs in each Industry type.

Industry	No. of IPOs	Percent
Computers	386	28.74
Electronic equipment	163	12.14
Scientific instrument	157	11.69
Chemical products	133	9.90
Retail	100	7.45
Manufacturing	49	3.65
Transportation	46	3.43
Health	40	2.98
Communications	14	1.04
Utilities	8	0.60
Financial services	5	0.37
Oil and Gas	4	0.30
All others	238	17.72
Total	1343	100.00

5.5 Results and Discussion

We use a smart method which is able to consider all kind of relations between data in addition to linear dependencies while capturing cross sectional effects. This method

automatically omits a irrelevant variable and removes its noise. We are interested in shed light on economic relationships like existing studies by some analyses of complex models like DNN exist in the literature and rating proxy variables and find the most impressive variables on each dependent variable of IPOs.

5.5.1 Estimation and Prediction of Components

In order to find the models between different factors and dependent variables and train DNN it is a common practice to divide all the observation into two parts. The training part of observation include 70% of IPO samples randomly and the rest are kept to test the power of the model in predictions. We compare our method with ordinary least square (OLS) which is the most used traditional technique in the literature. Table 5.3 show the performance of the DNN compared to OLS in predictions. Each model is first fitted on data to predict offer price (Panel A), and then after efficiency recovery and aftermarket mispricing estimation, another model is fitted to predict aftermarket component (Panel B). The results are reported for both train an test sets. Mean square error (MSE) as a measure which shows the fitness of the estimated model to data is used. DNN as a powerful model satisfies the expectations and indicates the has a better performance compared to OLS with lower MSE over test samples. The comparison between DNN as a model that captures highly nonlinear relationships and OLS we use more appropriate measure that tests the symmetric mean absolute percentage error (SMAPE) (Armstrong, 1985). This measure is based on relative error and calculated as follows:

$$SMAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{200 \cdot |y_i - \hat{y}_i|}{|y_i + \hat{y}_i|} \tag{5.16}$$

where y_i and $\hat{y}_i$ are the measured value and the foretasted value of a variable for ith observation and $i = 1, \ldots, N$. Regarding offer price and aftermarket mispricing prediction DNN shows a better accuracy according to train and test results. We also compute R-squared which measures the portion of variations which is captures by the

fitted model. It is observed that the model estimated by DNN replicates the response variables better that those of OLS. According to these three measures, DNN strongly outperform the traditional model of OLS. As a result, the estimated components obtained by DNN and the interpretation of the relationships between data found by DNN is more reliable that those realized by OLS. This is due to the power of DNN in dealing with high dimensional data where OLS fails to fit a suitable model caused by over-fitting issue.

Table 5.3: Comparison of OLS and DNN power in offer price (Panel A) and aftermarket mispricing prediction (Panel B). We assign 70% of IPOs randomly to the training samples and the remaining 30% of IPOs are used as test samples.

	DNN		OLS	
Panel A: Offer price prediction	Train	Test	Train	Test
Mean Squared Error	0.060	0.062	0.073	0.072
SMAPE (%)	8.348	8.499	9.090	8.975
R-squared	0.645	0.591	0.576	0.529
Panel B: Aftermarket mispricing prediction				
Mean Squared Error	20.923	14.466	28.253	18.064
SMAPE (%)	108.164	167.980	212.790	471.484
R-squared	0.644	0.587	0.496	0.440

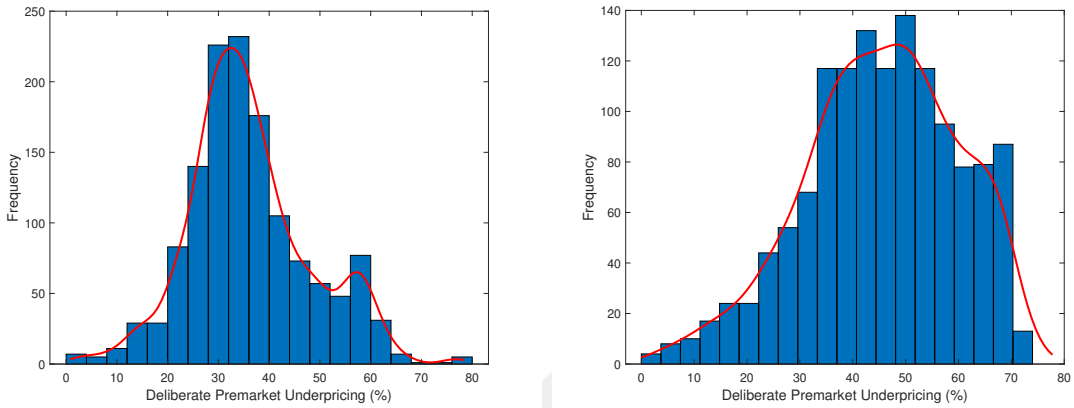
One of the most important outcomes of the suggested DNN approach is to compute the deliberate underpricing component of each IPO, while considering nonlinearities between data and also separating the interaction between premarket and aftermarket factors. We recover the efficiency or equivalently the level of deliberate premarket underpricing (DPU) for each IPO. This is computed for both methods of DNN and OLS. Table 5.4 shows the distribution of DPU over different intervals based on the percentage an IPO is underpriced in Panel A. DNN and OLS are close in terms of the number of IPOs for the level less than 20%. The main difference relates to the second and third intervals. The majority of IPOs (857 IPOs) indicate a DPU between 20-40%, however DNN suggests that most of IPOs are underpriced at a level between 40-60%.

For the forth interval DNN assigns more IPOs (237 IPOs) than OLS (45 IPOs). Both methods specify that no IPO is underpriced at a level more that 80%. In short, Panel B represent the statistics of DPU for both models. Evidently, there is a considerable shift equal to 9% in the mean of DPU obtained by DNN compared to OLS. Standard deviations are relatively close and a bit more in DNN. The negative skewness in DNN indicate that IPOs are prone to be underpriced at a higher percentage compared to OLS with positive skewness. Finally, larger kurtosis in OLS resulted from the basics in traditional models which are based on central moments and are not able to consider tails in sample observations. That is why the DPUs are centered around the mean in OLS and have larger kurtosis. These are major results that show how much it is crucial to capture all the relations and replicate observations which are rare events in the sample set, and how this consideration affect the knowledge extracted from data.

Table 5.4: Distribution and the statistics of the estimated deliberate premarket underpricing (DPU) by OLS and DNN.

Panel A:	DNN		OLS	
Distribution of DPU (%)	No. of firms	Percent	No. of firms	Percent
<20	68	5.06	81	6.03
20-40	397	29.56	857	63.81
40-60	641	47.71	360	26.80
60-80	237	17.64	45	3.35
80-100	0	0.00	0	0.00
Total	1343	100.00	1343	100.00
Panel B:				
Statistics of DPU (%)	Mean	StdDev	Skewness	Kurtosis
DNN	45.42	14.51	-0.36	2.74
OLS	36.14	12.05	0.41	3.36

The distribution of DPU is plotted for both DNN and OLS in figure 5.5. The notable point which is obvious in both figures is that the fitted kernel distribution on each histogram represents a bi-modal plot with a less frequent of larger mode level.



(a) COLS (b) DNN

Figure 5.5: The frequency histogram of deliberate premarket underpricing.

This result offers a mixture Gaussian distribution is suitable to model DPU with parametric methods.

5.5.2 Interpretation of Relations

It is a common practice in the literature to fit linear models on variables and draw conclusions based on the estimated coefficients. Cross sectional issue between high dimensional data leads to conflicting results of variables effect detection and economic interpretations. In previous section, we show that DNN effectively handles a broad range of complex relationships and addresses the cross sectional and over-fitting issues in the presence of high dimensional data. Consequently, the analyses obtained through DNN are verifiable and trusty.

Table 5.5 reports the coefficients and their statistics for OLS. Panel A is the result on offer price regression and Panel B includes those of aftermarket fitted line. Regarding the strength of various factors in pricing, NI, assets, leverage, fees, equity retained, proceeds, and underwriter reputation are significant at a confidence level more than 90%. Interestingly, all the variables considered as aftermarket factor are significant at a confidence level more than 95%.

Table 5.5: Regression results

Panel A: Regression of offer price on pricing and underpricing factors				
	Estimate	Std.Error	t-Stat	p-Value
(Intercept)	2.3983^{***}	0.0073	327.5495	0.0000
ln (EBITDA)	0.0024	0.0087	0.2693	0.7877
NI	-0.0173^{*}	0.0088	−1.9529	0.0510
R&D	−0.0014	0.0095	−0.1466	0.8835
ln (Assets)	0.0550^{***}	0.0132	4.1728	0.0000
ln (Leverage)	-0.0146^{*}	0.0081	−1.7964	0.0727
ln (Sales)	0.0101	0.0121	0.8342	0.4043
Industry dummy	0.0019	0.0090	0.2163	0.8288
Underwriter fees (%)	-0.0912^{***}	0.0086	−10.5773	0.0000
ln (Firm age)	0.0051	0.0078	0.6554	0.5123
Equity retained	0.0171^{**}	0.0077	2.2109	0.0272
ln (Proceeds)	0.1183^{***}	0.0090	13.1739	0.0000
Lock in	−0.0013	0.0078	−0.1700	0.8650
VC/PE backing	−0.0031	0.0088	−0.3550	0.7226
Underwriter reputation	0.1265^{***}	0.0096	13.1114	0.0000
Hot/Cold market	0.0108	0.0077	1.3934	0.1637
Hot/Cold underpricing	−0.0126	0.0079	−1.6012	0.1096
Hot/Cold demand for capital	−0.0041	0.0077	−0.5329	0.5942
R-squared	0.5665			
Panel B: Regression of aftermarket mispricing on aftermarket factors				
(Intercept)	3.6474^{***}	0.1366	26.7007	0.0000
Offer size	-1.5029^{***}	0.1493	−10.0687	0.0000
Shares traded	-3.5026^{***}	0.1534	−22.8347	0.0000
Above/Below range	-1.6033^{***}	0.1482	−10.8210	0.0000
Equity retained	-0.5326^{***}	0.1457	−3.6545	0.0003
Underwriter reputation	-0.3081^{**}	0.1515	−2.0329	0.0423
R-squared	0.4883			

Note. $^{*}p < 0.10$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Similar to any new statistical methods, structure of DNN is very complex and it is not straight froward to observe the coefficient of various inputs. DNN serves as a black box in which data have highly nonlinear interactions. There are several methods to study the contribution of variables in DNN. We implement two of them to investigate and determine the effect of each factor set on the associated response

variable. The first is Partial Derivative (PaD) method which computes the partial derivative of the DNN output with respect to the input (Dimopoulos et al., 1995; Gevrey et al., 2003). Let consider in each of the subnetworks, n_i denotes the number of inputs, n_h the number of hidden nodes in the hidden layer, and n_o be the number of outputs which is $n_o = 0$ in each subnetwork. The partial derivatives of each output observation $j \in \{1, \ldots, N\}$ with respect to input x_i with sigmoid function s in hidden nodes, are:

$$d_{ji} = S_j \sum_{h=1}^{n_h} w_{ho} s_{hj} (1 - s_{hj}) w_{ih} \tag{5.17}$$

where S_j is the derivative of the output node to its input which is one here as the simple addition function is used in output nodes. s_{hj} is the output of the hth hidden node, w_{ho} and w_{ih} are the weights of edges between hth hidden node and output node, and between ith input node and hth hidden node. This method results in obtaining a profile of output variations and helps to have a direct access to the influence of factors on the components e.g. positive or negative value for d_{ji} indicates a small change in an input factor causes the output variable decreases or increases.

The other result of PaD method is to identify the relative importance of each input variable and rank them based on the variation determination in total output variations. Sum of square derivatives (SSD) is calculated for each of input variable:

$$SSD_i = \sum_{j=1}^{N} (d_{ji})^2 \tag{5.18}$$

The inputs with higher SSDs are known as variales that have the most influences on the output variable.

The relative importance (RI) of factors in predictions are calculated in table 5.6 under column RI DNN and OLS method. We use regression coefficients in OLS and SSDs of DNN method to identify the most important input variables in descending order. We assume a value of RI=100 for the variable with largest linear coefficient

and SSD for OLS and DNN method respectively, and then compute RI for other inputs accordingly. The rsults show that underwriter fees and assets are the most contributing pricing factors in OLS approach. Concerning the underpricing factors, underwriter reputation and proceeds have the highest RI in DPU determination. In the aftermarket fitted line, shares traded has the largest RI and Above/below range and offer size contributed equally (about 46% and 43%, respectively) in aftermarket estimation. In DNN method, similar to OLS, underwriter fees has the highest RI but assets has the least influence in pricing. NI and sales are the second and third important factors on fair offer price. Proceeds by far is the most effective variable in DPU identified by DNN method. Lock in and VC/PE backing are at the bottom of the ranking list for both method. In the aftermarket, offer size ranked first for DNN while it is third for OLS.

The correlation of the pricing factors with offer price in OLS and DNN are mostly similar except for EBITDA. This factor has a positive affect on pricing in OLS while DNN finds a negative correlation. Among the underpricing factors, Hot/Cold demand for capital and VC/PE backing have negative relations with DPU by OLS method. IN DNN, these factors are positively correlated with DPU. The correlations between aftermarket factors and the recovered mispricing component are all nagative and the same for OLS and DNN methods.

Table 5.6: Relative importance and direction of each variable.

	Partial derivatives of DNN			Coefficient of OLS		
	Factor	RI	Correlation	Factor	RI	Correlation
Panel A:						
Pricing	Underwriter fees	100.00	-	Underwriter fees	100.00	-
	NI	38.90	-	Assets	60.30	+
	Sales	9.70	+	NI	18.95	-
	Leverage	2.90	-	Leverage	15.96	-
	EBITDA	1.51	-	Sales	11.06	+
	Industry dummy	0.81	+	EBITDA	2.58	+
	R&D	0.28	-	Industry dummy	2.12	+
	Assets	0.03	+	R&D	1.52	-
Panel B:						
Premarket underpricing	Proceeds	100.00	+	Underwriter reputation	100.00	+
	Underwriter reputation	8.72	+	Proceeds	93.52	+
	Equity retained	0.66	+	Equity retained	13.49	+
	Hot/Cold demand for capital	0.63	+	Hot/Cold underpricing	9.98	-
	Hot/Cold market	0.24	+	Hot/Cold market	8.51	+
	Firm age	0.17	+	Firm age	4.05	+
	Hot/Cold underpricing	0.08	-	Hot/Cold demand for capital	3.26	-
	VC/PE backing	0.01	+	VC/PE backing	2.47	-
	Lock in	0.00	-	Lock in	1.05	-
Panel C:						
Aftermarket mispricing	Offer size	100.00	-	Shares traded	100.00	-
	Shares traded	26.20	-	Above/Below range	45.77	-
	Above/Below range	5.40	-	Offer size	42.91	-
	Equity retained	2.72	-	Equity retained	15.20	-
	Underwriter reputation	1.07	-	Underwriter reputation	8.79	-

We implement another analysis on the network sensitivities. Perturb method is based on observing the effect of small changes in inputs on the output of the network. The algorithm alters one input variable by some specified amounts while keeping all the other inputs at their main values (Gevrey et al., 2003).

Figure 5.6 shows the profile of variations in each set of factors. We perturb each input from -50% to +50% of its initial value. Three results are assessed by plotting the input changes against the changes in response variable. First, the input which affects the output more has more relative importance. This finding is consistent with PaD method and RIs reported in table 5.6 Second, the change in output is measured by mean squared error which increases as more noise is added to an input. That is the reason for the curve plotted to be in a convex form an a minimum value at initial value of inputs or equivalently no changes in them. Third, in figure 5.6(a), the positive and negative perturbations have symmetric influence on output variations. However, this result is different for underpricing and aftermarket mispricing analyses in figures 5.6(b) and 5.6(c), respectively. For example, less amounts of proceeds affect the level of underpricing more than higher values for proceeds. Similar to this result is also observed for all the aftermarket factors on the mispricing response variable.

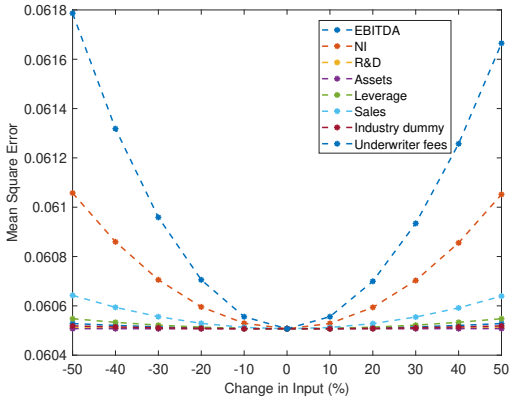
5.6 Conclusion

The main contributions and findings of this chapter are as follows:

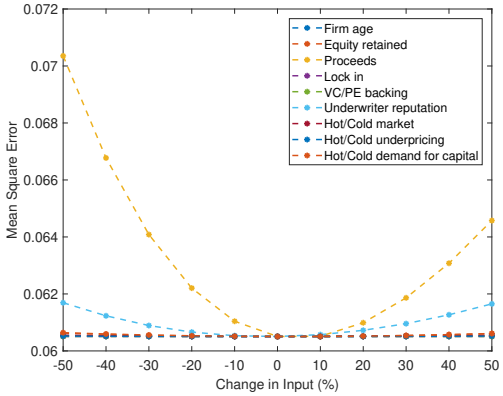
- A nonlinear SFA approach based on machine learning is proposed to estimate the intrinsic value of a firm before going public.

- The proposed distribution-free model allows the user to estimate the premarket underpricing simply, without assuming any distribution for it.

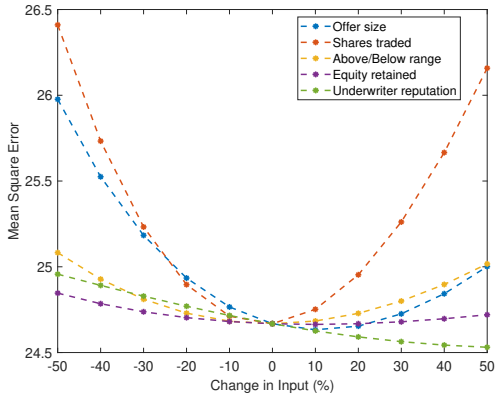
- The empirical application of the model provides evidence that only a few determinants of the value of firms and deal characteristics impact the pricing and



(a) Pricing variables



(b) Underpricing variables



(c) Mispricing variables

Figure 5.6: Sensitivity analysis of different factors on network's output.

underpricing of IPOs when we take into account nonlinear interaction among the variables.

Chapter 6

CONCLUSION

This thesis investigates and develops data-driven methods using machine learning techniques in some problems where decisions need to be made under a stochastic environment in operations and finance. Data-driven methods have several advantages that include reducing the statistical and mathematical assumptions, depending on the decisions directly on available data, and integrating estimation and optimization as two main steps in stochastic optimization problems. We propose to develop this approach using a neural network as a powerful method in function approximation and modeling complex relations between statistical information.

In inventory problems, feature data about the demand for a product and underlying dynamics of the market that have often dependencies across multiple decision periods are both common factors in the ordering decision. We suggest a novel approach in data-driven, which considers both observable and unobservable sources of features that affect the randomness in demand. This new model extends the existing data-driven methods to other applications, where there are limitations in identifying the factors and their volatility that influence the state of the system. We then incorporate the hidden Markov model as the most used modeling of the evolving environment into the data-driven model. Utilizing HMM in a data-driven framework enables us to model long-term dependencies and take advantage of other information sources available in the form of the feature data.

By combining neural network modeling tools with stochastic inference, and integrating them into the optimization method, we propose an integrated solution method for the suggested model that captures nonlinear dependencies between features and order quantity while alleviating the errors that occur in the estimation step. Finally,

we show the robustness of our proposed solution method using extensive numerical experiments with both synthetic and real data. We compare the results of the suggested model with data-driven methods that ignore the evolving environment as a hidden factor or that employ inference and optimization steps separately. Our numerical results reveal that the proposed approach performs better than other methods when there might be unobservable features and leads to the recommendation that taking into account the unobservable features might have significant benefits. In addition, we present evidence that integrating unobservable feature estimation and inventory optimization is feasible and may bring significant improvements.

In asset allocation, we suggest a novel approach to conditional asset allocation using neural networks, which builds a complex model using state variables to generate optimal portfolio weights directly and without estimation of the return distribution or fitting a model to data. The approach captures nonlinear dependencies between state variables and portfolio weights. Using neural networks makes it possible for the investor to input a range of possible state variables, regardless of their explanatory power on return prediction. Networks distinguish ineffective inputs and eliminate their additional noise. We show the robustness of our proposed DNN-based method using empirical data and various performance ratios. In addition, we introduce three analytical methods adopted from computer science that help to interpret the outcome of any machine learning techniques used in finance and economics by analyzing the influence of input variables on the response variable. This is particularly important, as machine learning tools are criticized in finance and economics fields as being a black box that cannot be used for economic interpretation. We use these analytical methods and show the relative impact of trend and dividend yield on the Sharpe ratio, as the most important performance ratio used by practitioners in portfolio advice. We also provide evidence that shows that the magnitude and direction of the impact of the state variables in determining the Sharpe ratio vary over time. These results suggest that practitioners who use the Sharpe ratio should pay attention to these state variables and the sensitivity of their portfolios to the (sometimes mixed)

signals that these variables send. The important implication of these findings is that investors should not rely on a single or a specific combination of state variables in their conditional asset allocations since their predictability varies over time.

Regarding a state-dependent investor's risk profile, we consider a one-period asset allocation in a myopic strategy where the portfolio weights do not change during each period. Rather we rebalance optimally by dynamically switching to the best performance measures at the beginning of each period. This allows us to make correctly risk-adjusted portfolio choices during an investment horizon. Neural networks are capable of estimating the best risk profile depending on the state of the market and take advantage of the investment opportunities.

In this thesis, we also use machine learning tools to offer price prediction and efficiency estimation in IPO were many factor data have complicated relations. This multi-dimensional data includes similarities with cross-sectional issues. New statistical methods can model high order data concerning the number of features and consider a wider range of data points that are not modeled efficiently by traditional methods.

In the prediction part, we develop and use a Deep Neural Network (DNN) with three sets of input variables proposed to predict the offer prices and first-day initial returns of IPOs, while obtaining an estimation for deliberate underpricing. This approach provides a nonparametric estimation for deliberate underpricing. The proposed method follows the main rules of SFA. Hence, we look for a maximum value as a frontier, while a one-sided variable representing deliberate underpricing and a normal error reduce the fair price to the offer price. We also relax the distributional assumption of the deliberate underpricing component. Moreover, this method considers complex relationships between the aftermarket mispricing component and its associated proxies. This method benefits from a deliberate underpricing component that originates from the premarket and partially forms the first day's initial return. It can estimate the fair price, deliberate underpricing, and aftermarket mispricing. However, for a new IPO, we pick a part of the model to estimate the deliberate underpricing with no need for aftermarket information. In summary, a newly designed

DNN connects a set of proxies to two dependent variables representing the offer price and initial return by using the potentially deliberate underpricing as an unknown but key component.

We apply the proposed approach in the U.S. IPO market and use two methods from computer science for analyzing the complex estimated models and draw economical interpretations for the impact of independent variables on dependent variables. The analyses of our estimated models reveal several remarkable points about the importance of variables and correlations. We find that only a few determinants of the value of firms impact the pricing and underpricing of IPOs. The underwriter fee and negative income play the most important role in IPO price efficiency, among various pricing variables. Proceeds followed by underwriter reputation significantly impact the premarket underpricing, while VC/PE backing, and lock-in agreement are not associated with the premarket inefficiency. We estimate aftermarket mispricing and show that it is attributed more to offer size and shares traded, and the underwriter's reputation is not an important determinant of aftermarket mispricing. Further, we find that RD expenses and assets have a relatively weak relation with IPO efficiency. We show that proceeds have a positive relation with deliberate underpricing. Our results indicate that while a hot market, as well as hot demand for capital, has a positive, important impact on premarket underpricing, hot/cold underpricing cycles do not play a significant role in explaining premarket underpricing.

The studies in this thesis can be extended in different directions. Although we consider only the ordering problem for a single-item newsvendor in the inventory system, our approach can be extended to multiple items whose demands could be correlated. Another interesting extension is that of a joint price and inventory optimization problem where the demand is dependent on the selling price. In this case, one can investigate how to switch between pricing strategies during various hidden states and use environmental and local features as price drivers.

In addition to modeling non-linear relations between input and optimal targets and decisions, neural networks are capable of taking many input data, ranging from

time series to textual news in social media as additional databases. In asset allocation, we use microeconomic variables as state variables, an extension to this work would be using asset classes' level or firms' level characteristics as inputs that involve many input variables. Additionally, neural networks can provide more outputs for portfolios consisting of more stocks such as exchange-traded funds (ETF). The number of output nodes increases to the number of stocks from which the corresponding weight in the portfolio is obtained. This provides a research direction to utilize the power of neural networks in considering many input variables.

BIBLIOGRAPHY

Adcock, C., Areal, N., Cortez, M. C., Oliveira, B., and Silva, F. (2019). Portfolio performance persistence: does the choice of performance measure matter?

Agarwal, V. and Naik, N. Y. (2004). Risks and portfolio decisions involving hedge funds. *The Review of Financial Studies*, 17(1):63–98.

Aggarwal, R., Bhagat, S., and Rangan, S. (2009). The impact of fundamentals on ipo valuation. *Financial Management*, 38(2):253–284.

Aggarwal, R. and Rivoli, P. (1990). Fads in the initial public offering market? *Financial Management*, pages 45–57.

Aigner, D., Lovell, C. K., and Schmidt, P. (1977). Formulation and estimation of stochastic frontier production function models. *Journal of econometrics*, 6(1):21–37.

Aït-Sahalia, Y. and Brandt, M. W. (2001). Variable selection for portfolio choice. *The Journal of Finance*, 56(4):1297–1351.

Akerlof, G. A. (1978). The market for "lemons": Quality uncertainty and the market mechanism. In *Uncertainty in Economics*, pages 235–251. Elsevier.

Allen, F. and Faulhaber, G. R. (1989). Signalling by underpricing in the ipo market. *Journal of financial Economics*, 23(2):303–323.

Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.

Ang, A. and Bekaert, G. (2004). How regimes affect asset allocation. *Financial Analysts Journal*, 60(2):86–99.

Arba, M. B. and Karaa, A. (2006). Ipo's initial returns: Underpricing versus noisy trading. *EURO-MEDITERRANEAN ECONOMICS AND FINANCE*, page 48.

Arifoğlu, K. and Özekici, S. (2010). Optimal policies for inventory systems with finite capacity and partially observed markov-modulated demand and supply processes. *European Journal of Operational Research*, 204(3):421–438.

Arifoğlu, K. and Özekici, S. (2011). Inventory management with random supply and imperfect information: A hidden markov model. *International Journal of Production Economics*, 134(1):123–137.

Armstrong, J. S. (1985). Long-range forecasting: From crystal ball to computer. *New York ua*.

Arrow, K. J., Harris, T., and Marschak, J. (1951). Optimal inventory policy. *Econometrica: Journal of the Econometric Society*, pages 250–272.

Arthurs, J. D., Busenitz, L. W., Hoskisson, R. E., and Johnson, R. A. (2009). Signaling and initial public offerings: The use and impact of the lockup period. *Journal of Business Venturing*, 24(4):360–372.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999). Coherent measures of risk. *Mathematical finance*, 9(3):203–228.

Avci, H., Gokbayrak, K., and Nadar, E. (2020). Structural results for average-cost inventory models with markov-modulated demand and partial information. *Production and Operations Management*, 29(1):156–173.

Ban, G.-Y. and Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108.

Barberis, N. (2000). Investing for the long run when returns are predictable. *The Journal of Finance*, 55(1):225–264.

Baron, D. P. (1982). A model of the demand for investment banking advising and distribution services for new issues. *The Journal of Finance*, 37(4):955–976.

Barry, C. B., Muscarella, C. J., Peavy Iii, J. W., and Vetsuypens, M. R. (1990). The role of venture capital in the creation of public companies: Evidence from the going-public process. *Journal of Financial economics*, 27(2):447–471.

Bartling, B. and Park, A. (2009). What determines the level of ipo gross spreads? underwriter profits and the cost of going public. *International Review of Economics & Finance*, 18(1):81–109.

Battese, G. E. and Coelli, T. J. (1995). A model for technical inefficiency effects in a stochastic frontier production function for panel data. *Empirical economics*, 20(2):325–332.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (2013). *Nonlinear programming: theory and algorithms*. John Wiley & Sons.

Beatty, R. P. and Ritter, J. R. (1986). Investment banking, reputation, and the underpricing of initial public offerings. *Journal of financial economics*, 15(1-2):213–232.

Bekaert, G. and Hodrick, R. J. (1992). Characterizing predictable components in excess returns on equity and foreign exchange markets. *The Journal of Finance*, 47(2):467–509.

Bensoussan, A., Cakanyıldırım, M., and Sethi, S. P. (2005). On the optimal control of partially observed inventory systems. *Comptes Rendus Mathematique*, 341(7):419–426.

Bensoussan, A., Çakanyıldırım, M., and Sethi, S. P. (2007). A multiperiod newsvendor problem with partially observed demand. *Mathematics of Operations Research*, 32(2):322–344.

Bernard, C., Chen, J. S., and Vanduffel, S. (2015a). Rationalizing investors' choices. *Journal of Mathematical Economics*, 59:10–23.

Bernard, C., Moraux, F., Rüschendorf, L., and Vanduffel, S. (2015b). Optimal payoffs under state-dependent preferences. *Quantitative Finance*, 15(7):1157–1173.

Bernard, C., Vanduffel, S., and Ye, J. (2018). Optimal portfolio under state-dependent expected utility. *International Journal of Theoretical and Applied Finance*, 21(03):1850013.

Bertsimas, D. and Kallus, N. (2014). From predictive to prescriptive analytics. *arXiv preprint arXiv:1402.5481*.

Bertsimas, D. and Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044.

Bertsimas, D. and Thiele, A. (2005). A data-driven approach to newsvendor problems. *Working Paper, Massachusetts Institute of Technology*.

Besbes, O. and Muharremoglu, A. (2013). On implications of demand censoring in the newsvendor problem. *Management Science*, 59(6):1407–1424.

Besiou, M. and Van Wassenhove, L. N. (2015). Addressing the challenge of modeling for decision-making in socially responsible operations. *Production and Operations Management*, 24(9):1390–1401.

Beyer, D. and Sethi, S. P. (1997). Average cost optimality in inventory models with markovian demands. *Journal of Optimization Theory and Applications*, 92(3):497–526.

Biglova, A., Ortobelli, S., Rachev, S. T., and Stoyanov, S. (2004). Different approaches to risk estimation in portfolio theory. *The Journal of Portfolio Management*, 31(1):103–112.

Billio, M., Caporin, M., and Costola, M. (2015). Backward/forward optimal combination of performance measures for equity screening. *The North American Journal of Economics and Finance*, 34:63–83.

Blinder, A. S. and Maccini, L. J. (1991). The resurgence of inventory research: what have we learned? *Journal of Economic Surveys*, 5(4):291–328.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327.

Bollerslev, T., Engle, R. F., and Wooldridge, J. M. (1988). A capital asset pricing model with time-varying covariances. *Journal of political Economy*, 96(1):116–131.

Bradrania, R. and Neghab, D. P. (2021). State-dependent asset allocation using neural networks. *Revised and resubmitted to European Journal of Finance*.

Brailsford, T., Heaney, R., and Shi, J. (2004). Modelling the behaviour of the new issue market. *International review of financial analysis*, 13(2):119–132.

Brandt, M. W. and Santa-Clara, P. (2006). Dynamic portfolio selection by augmenting the asset space. *The Journal of Finance*, 61(5):2187–2217.

Brandt, M. W., Santa-Clara, P., and Valkanov, R. (2009). Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns. *The Review of Financial Studies*, 22(9):3411–3447.

Brav, A. and Gompers, P. A. (1997). Myth or reality? the long-run underperformance of initial public offerings: Evidence from venture and nonventure capital-backed companies. *The Journal of Finance*, 52(5):1791–1821.

Brav, A. and Gompers, P. A. (2003). The role of lockups in initial public offerings. *The Review of Financial Studies*, 16(1):1–29.

Brennan, M. J., Schwartz, E., and Lagnado, R. (1998). The use of treasury bill futures in strategic asset allocation programs. *Worldwide Asset and Liability Modeling, Cambridge University Press, Cambridge, UK*, pages 205–228.

Brennan, M. J., Schwartz, E. S., and Lagnado, R. (1997). Strategic asset allocation. *Journal of Economic Dynamics and Control*, 21(8-9):1377–1403.

Butler, J. and Schachter, B. (1997). Estimating value-at-risk with a precision measure by combining kernel estimation with historical simulation. *Review of Derivatives Research*, 1:371–390.

Campbell, J. Y. (1987). Stock returns and the term structure. *Journal of financial economics*, 18(2):373–399.

Campbell, J. Y., Lewis Chanb, Y., and Viceira, M. (2013). A multivariate model of strategic asset allocation. In *HANDBOOK OF THE FUNDAMENTALS OF FINANCIAL DECISION MAKING: Part II*, pages 809–848. World Scientific.

Campbell, J. Y. and Viceira, L. M. (1999). Consumption and portfolio decisions when expected returns are time varying. *The Quarterly Journal of Economics*, 114(2):433–495.

Campbell, J. Y., Viceira, L. M., Viceira, L. M., et al. (2002). *Strategic asset allocation: portfolio choice for long-term investors*. Clarendon Lectures in Economic.

Campbell, R., Huisman, R., and Koedijk, K. (2001). Optimal portfolio selection in a value-at-risk framework. *Journal of Banking & Finance*, 25(9):1789–1804.

Caporin, M. and Lisi, F. (2011). Comparing and selecting performance measures using rank correlations. *Economics Discussion Papers*, (14).

Carter, R. and Manaster, S. (1990). Initial public offerings and underwriter reputation. *the Journal of Finance*, 45(4):1045–1067.

Carter, R. B. (1992). Underwriter reputation and repetitive public offerings. *Journal of Financial Research*, 15(4):341–354.

Chahine, S. (2007). Investor interest, trading volume, and the choice of ipo mechanism in france. *International Review of Financial Analysis*, 16(2):116–135.

Chan, L. K., Karceski, J., and Lakonishok, J. (1999). On portfolio optimization: Forecasting covariances and choosing the risk model. *The review of Financial studies*, 12(5):937–974.

Chang, C.-L., McAleer, M., and Wong, W.-K. (2018). Big data, computational science, economics, finance, marketing, management, and psychology: connections. *Journal of Risk and Financial Management*, 11(1):15.

Chen, A., Hung, C. C., and Wu, C.-S. (2002). The underpricing and excess returns of initial public offerings in taiwan based on noisy trading: a stochastic frontier model. *Review of Quantitative Finance and Accounting*, 18(2):139–159.

Chen, H.-H., Tsai, H.-T., and Lin, D. K. (2011). Optimal mean-variance portfolio selection using cauchy–schwarz maximization. *Applied economics*, 43(21):2795–2801.

Chen, S.-S. (2009). Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance*, 33(2):211–223.

Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., and De Felice, F. (2020). Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. *Sustainability*, 12(2):492.

Cogneau, P. and Hübner, G. (2009). The 101 ways to measure portfolio performance. *Available at SSRN 1326076*.

Colombo, E., Forte, G., and Rossignoli, R. (2017). Carry trade returns with support vector machines. *International Review of Finance*.

Costa, G. and Kwon, R. H. (2019). Risk parity portfolio optimization under a markov regime-switching framework. *Quantitative Finance*, 19(3):453–471.

Cox, J. C., Ingersoll Jr, J. E., and Ross, S. A. (1985). An intertemporal general equilibrium model of asset prices. *Econometrica: Journal of the Econometric Society*, pages 363–384.

Deeds, D. L., Decarolis, D., and Coombs, J. E. (1997). The impact of firmspecific capabilities on the amount of capital raised in an initial public offering: Evidence from the biotechnology industry. *Journal of Business venturing*, 12(1):31–46.

DeMiguel, V., Garlappi, L., Nogales, F. J., and Uppal, R. (2009). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management science*, 55(5):798–812.

Diebold, F. X., Lee, J.-H., and Weinbach, G. C. (1994). Regime switching with time-varying transition probabilities. *Business Cycles: Durations, Dynamics, and Forecasting*, 1:144–165.

Dimopoulos, Y., Bourret, P., and Lek, S. (1995). Use of some sensitivity criteria for choosing networks with good generalization ability. *Neural Processing Letters*, 2(6):1–4.

Efendigil, T., Önüt, S., and Kahraman, C. (2009). A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis. *Expert Systems with Applications*, 36(3):6697–6707.

Ekholm, A. and Pasternack, D. (2005). The negative news threshold—an explanation for negative skewness in stock returns. *The european journal of finance*, 11(6):511–529.

Ellis, K. (2006). Who trades ipos? a close look at the first days of trading. *Journal of Financial Economics*, 79(2):339–363.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007.

Ernst, C., Grossmann, M., Hocht, S., Minden, S., Scherer, M., Zagst, R., et al. (2009). Portfolio selection under changing market conditions. *International Journal of Financial Services Technology Management*, 4(1):48.

Esfahanipour, A., Goodarzi, M., and Jahanbin, R. (2016). Analysis and forecasting of ipo underpricing. *Neural Computing and Applications*, 27(3):651–658.

Fama, E. F. and French, K. R. (1988). Dividend yields and expected stock returns. *Journal of financial economics*, 22(1):3–25.

Fama, E. F. and French, K. R. (1989). Business conditions and expected returns on stocks and bonds. *Journal of financial economics*, 25(1):23–49.

Fama, E. F. and French, K. R. (1997). Industry costs of equity. *Journal of financial economics*, 43(2):153–193.

Fama, E. F. and Schwert, G. W. (1977). Asset returns and inflation. *Journal of financial economics*, 5(2):115–146.

Farinelli, S., Ferreira, M., Rossello, D., Thoeny, M., and Tibiletti, L. (2008). Beyond sharpe ratio: Optimal asset allocation using different performance ratios. *Journal of Banking & Finance*, 32(10):2057–2063.

Farrell, M. J. (1957). The measurement of productive efficiency. *Journal of the Royal Statistical Society: Series A (General)*, 120(3):253–281.

Feldman, R. M. (1978). A continuous review (s, s) inventory system in a random environment. *Journal of Applied Probability*, 15(3):654–659.

Ferson, W. E. and Harvey, C. R. (1991). The variation of economic risk premiums. *Journal of Political Economy*, 99(2):385–415.

Fishburn, P. C. (1977). Mean-risk analysis with risk associated with below-target returns. *The American Economic Review*, 67(2):116–126.

Francis, B. B. and Hasan, I. (2001). The underpricing of venture and nonventure capital ipos: An empirical investigation. *Journal of Financial Services Research*, 19(2-3):99–113.

French, K. R., Schwert, G. W., and Stambaugh, R. F. (1987). Expected stock returns and volatility. *Journal of financial Economics*, 19(1):3–29.

Friesen, G. C. and Swift, C. (2009). Overreaction in the thrift ipo aftermarket. *Journal of Banking & Finance*, 33(7):1285–1298.

Gallego, G. and Hu, H. (2004). Optimal policies for production/inventory systems with finite capacity and markov-modulated demand and supply processes. *Annals of Operations Research*, 126(1-4):21–41.

Gallego, G. and Moon, I. (1993). The distribution free newsboy problem: review and extensions. *Journal of the Operational Research Society*, 44(8):825–834.

Gevrey, M., Dimopoulos, I., and Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, 160(3):249–264.

Goel, S., Hofman, J. M., Lahaie, S., Pennock, D. M., and Watts, D. J. (2010). Predicting consumer behavior with web search. *Proceedings of the National academy of sciences*.

Gompers, P. A. (1996). Grandstanding in the venture capital industry. *Journal of Financial economics*, 42(1):133–156.

Gordon, S. and St-Amour, P. (2000). A preference regime model of bull and bear markets. *American Economic Review*, 90(4):1019–1033.

Gordon, S. and St-Amour, P. (2004). Asset returns and state-dependent risk preferences. *Journal of Business & Economic Statistics*, 22(3):241–252.

Greene, W. (2005). Fixed and random effects in stochastic frontier models. *Journal of productivity analysis*, 23(1):7–32.

Greene, W. H. (1990). A gamma-distributed stochastic frontier model. *Journal of econometrics*, 46(1-2):141–163.

Griffin, J. E. and Steel, M. F. (2008). Flexible mixture modelling of stochastic frontiers. *Journal of Productivity Analysis*, 29(1):33–50.

Grinblatt, M. and Hwang, C. Y. (1989). Signalling and the pricing of new issues. *The Journal of Finance*, 44(2):393–420.

Gruhl, D., Chavet, L., Gibson, D., Meyer, J., Pattanayak, P., Tomkins, A., and Zien, J. (2004). How to build a webfountain: An architecture for very large-scale text analytics. *IBM Systems Journal*, 43(1):64–77.

Gruhl, D., Guha, R., Kumar, R., Novak, J., and Tomkins, A. (2005). The predictive power of online chatter. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 78–87. ACM.

Guidolin, M. and Timmermann, A. (2007). Asset allocation under multivariate regime switching. *Journal of Economic Dynamics and Control*, 31(11):3503–3544.

Hannah, L., Powell, W., and Blei, D. M. (2010). Nonparametric density estimation for stochastic optimization with an observable state variable. In *Advances in Neural Information Processing Systems*, pages 820–828.

Harvey, C. R. (2001). The specification of conditional expectations. *Journal of Empirical Finance*, 8(5):573–637.

He, B., Dexter, F., Macario, A., and Zenios, S. (2012). The timing of staffing decisions in hospital operating rooms: incorporating workload heterogeneity into the newsvendor problem. *Manufacturing & Service Operations Management*, 14(1):99–114.

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

Hertzel, M. G., Huson, M. R., and Parrino, R. (2012). Public market staging: The timing of capital infusions in newly public firms. *Journal of Financial Economics*, 106(1):72–90.

Huang, G.-B. (2003). Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, 14(2):274–281.

Hughes, P. J. (1986). Signalling by direct disclosure under asymmetric information. *Journal of accounting and economics*, 8(2):119–142.

Huh, W. T., Levi, R., Rusmevichientong, P., and Orlin, J. B. (2011). Adaptive data-driven inventory control with censored demand based on kaplan-meier estimator. *Operations Research*, 59(4):929–941.

Hunt-McCool, J., Koh, S. C., and Francis, B. B. (1996). Testing for deliberate underpricing in the ipo premarket: A stochastic frontier approach. *The Review of Financial Studies*, 9(4):1251–1269.

Ibbotson, R. G., Sindelar, J. L., and Ritter, J. R. (1988). Initial public offerings. *Journal of applied corporate finance*, 1(2):37–45.

Jain, B. A. and Nag, B. N. (1995). Artificial neural network models for pricing initial public offerings. *Decision Sciences*, 26(3):283–302.

Jarque, C. M. and Bera, A. K. (1987). A test for normality of observations and regression residuals. *International Statistical Review/Revue Internationale de Statistique*, pages 163–172.

Kaastra, I. and Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3):215–236.

Kanas, A. (2008). A multivariate regime switching approach to the relation between the stock market, the interest rate and output. *International Journal of Theoretical and Applied Finance*, 11(07):657–671.

Kat, H. M. and Brooks, C. (2001). The statistical properties of hedge fund index returns and their implications for investors.

Ke, J. and Liu, X. (2008). Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction. In *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on*, volume 2, pages 828–832. IEEE.

Keim, D. B. and Stambaugh, R. F. (1986). Predicting returns in the stock and bond markets. *Journal of financial Economics*, 17(2):357–390.

Kesavan, S. and Kushwaha, T. (2014). Differences in retail inventory investment behavior during macroeconomic shocks: Role of service level. *Production and Operations Management*, 23(12):2118–2136.

Khayyati, S. and Tan, B. (2020). Data-driven control of a production system by using marking-dependent threshold policy. *International Journal of Production Economics*, 226:107607.

Konno, H. and Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. *Management science*, 37(5):519–531.

Koop, G. and Li, K. (2001). The valuation of ipo and seo firms. *Journal of Empirical Finance*, 8(4):375–401.

Krishnan, C., Ivanov, V. I., Masulis, R. W., and Singh, A. K. (2011). Venture capital reputation, post-ipo performance, and corporate governance. *Journal of Financial and Quantitative Analysis*, 46(5):1295–1333.

Kritzman, M., Page, S., and Turkington, D. (2012). Regime shifts: Implications for dynamic strategies (corrected). *Financial Analysts Journal*, 68(3):22–39.

Kroll, Y., Levy, H., and Markowitz, H. M. (1984). Mean-variance versus direct utility maximization. *The Journal of Finance*, 39(1):47–61.

Laborda, R. and Olmo, J. (2017). Optimal asset allocation for strategic investors. *International Journal of Forecasting*, 33(4):970–987.

Lee, Y. H. and Schmidt, P. (1993). A production frontier model with flexible temporal variation in technical efficiency. *The measurement of productive efficiency: Techniques and applications*, pages 237–255.

Leland, H. E. (1999). Beyond mean-variance: Performance measurement in a nonsymmetrical world. *Financial analysts journal*, pages 27–36.

Leone, A. J., Rock, S., and Willenborg, M. (2007). Disclosure of intended use of proceeds and underpricing in initial public offerings. *Journal of Accounting Research*, 45(1):111–153.

Lerner, J. (1994). Venture capitalists and the decision to go public. *Journal of financial Economics*, 35(3):293–316.

Levi, R., Roundy, R. O., and Shmoys, D. B. (2007). Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research*, 32(4):821–839.

Levinson, S. E., Rabiner, L. R., and Sondhi, M. M. (1983). An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell System Technical Journal*, 62(4):1035–1074.

Li, B. and Pi, D. (2019). Learning deep neural networks for node classification. *Expert Systems with Applications*, 137:324–334.

Liu, X. and Ritter, J. R. (2011). Local underwriter oligopolies and ipo underpricing. *Journal of Financial Economics*, 102(3):579–601.

Liyanage, L. H. and Shanthikumar, J. G. (2005). A practical inventory control policy using operational statistics. *Operations Research Letters*, 33(4):341–348.

Ljungqvist, A. and Wilhelm Jr, W. J. (2003). Ipo pricing in the dot-com bubble. *The Journal of Finance*, 58(2):723–752.

Loughran, T. and Ritter, J. (2004). Why has ipo underpricing changed over time? *Financial management*, pages 5–37.

Lovejoy, W. S. (1992). Stopped myopic policies in some inventory models with generalized demand processes. *Management Science*, 38(5):688–707.

Lowry, M. (2003). Why does ipo volume fluctuate so much? *Journal of Financial economics*, 67(1):3–40.

Lowry, M. and Schwert, G. W. (2002). Ipo market cycles: Bubbles or sequential learning? *The Journal of Finance*, 57(3):1171–1200.

Luo, C. and Ouyang, Z. (2014). Estimating ipo pricing efficiency by bayesian stochastic frontier analysis: The chinext market case. *Economic Modelling*, 40:152–157.

Luque, C., Quintana, D., Valls, J. M., and Isasi, P. (2009). Two-layered evolutionary forecasting for ipo underpricing. In *2009 IEEE Congress on Evolutionary Computation*, pages 2374–2378. IEEE.

Malandri, L., Xing, F. Z., Orsenigo, C., Vercellis, C., and Cambria, E. (2018). Public mood–driven asset allocation: The importance of financial sentiment in portfolio management. *Cognitive Computation*, 10(6):1167–1176.

Malkiel, B. G. and Saha, A. (2005). Hedge funds: Risk and return. *Financial analysts journal*, pages 80–88.

Markowitz, H. (1952). Portfolio selection. *The journal of finance*, 7(1):77–91.

Martin, R. D., Rachev, S. Z., and Siboulet, F. (2003). Phi-alpha optimal portfolios and extreme risk management. *The Best of Wilmott 1: Incorporating the Quantitative Finance Review*, 1:223.

Meeusen, W. and van Den Broeck, J. (1977). Efficiency estimation from cobb-douglas production functions with composed error. *International economic review*, pages 435–444.

Megginson, W. L. and Weiss, K. A. (1991). Venture capitalist certification in initial public offerings. *The Journal of Finance*, 46(3):879–903.

Merton, R. C. (1980). On estimating the expected return on the market: An exploratory investigation. *Journal of financial economics*, 8(4):323–361.

Merton, R. C. et al. (1973). An intertemporal capital asset pricing model. *Econometrica*, 41(5):867–887.

Miller, R. E. and Reilly, F. K. (1987). An examination of mispricing, returns, and uncertainty for initial public offerings. *Financial Management*, pages 33–38.

Monahan, G. E. (1982). State of the art—a survey of partially observable markov decision processes: theory, models, and algorithms. *Management science*, 28(1):1–16.

Moro Visconti, R. and Morea, D. (2019). Big data for the sustainability of healthcare project financing. *Sustainability*, 11(13):3748.

Nahata, R. (2008). Venture capital reputation and investment performance. *Journal of Financial Economics*, 90(2):127–151.

Nanda, R. and Rhodes-Kropf, M. (2013). Investment cycles and startup innovation. *Journal of Financial Economics*, 110(2):403–418.

Neghab, D. P., Bradrania, R., and Elliott, R. (2021a). Ipo pricing efficiency using machine learning: estimation and interpretation. *Submitted to Economic Modelling*.

Neghab, D. P., Khayyati, S., and Karaesmen, F. (2021b). An integrated data-driven method using deep learning for a newsvendor problem with unobservable features. *Submitted to European Journal of Operational Reseach*.

Nystrup, P., Madsen, H., and Lindström, E. (2018). Dynamic portfolio optimization across hidden market regimes. *Quantitative Finance*, 18(1):83–95.

Olden, J. D. and Jackson, D. A. (2002). Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, 154(1-2):135–150.

Oroojlooyjadid, A., Snyder, L. V., and Takáč, M. (2020). Applying deep learning to the newsvendor problem. *IISE Transactions*, 52(4):444–463.

Ortobelli, S., Kouaissah, N., and Tichỳ, T. (2017). On the impact of conditional expectation estimators in portfolio theory. *Computational Management Science*, 14(4):535–557.

Ortobelli, S., Vitali, S., Cassader, M., and Tichỳ, T. (2018). Portfolio selection strategy for fixed income markets with immunization on average. *Annals of Operations Research*, 260(1-2):395–415.

Peng, Y. and Wang, K. (2007). Ipo underpricing and flotation methods in taiwan–a stochastic frontier approach. *Applied Economics*, 39(21):2785–2796.

Perakis, G. and Roels, G. (2008). Regret in the newsvendor model with partial information. *Operations Research*, 56(1):188–203.

Pitt, M. M. and Lee, L.-F. (1981). The measurement and sources of technical inefficiency in the indonesian weaving industry. *Journal of development economics*, 9(1):43–64.

Poritz, A. (1982). Linear predictive hidden markov models and the speech signal. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.*, volume 7, pages 1291–1294. IEEE.

Provost, F. and Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1):51–59.

Qi, M., Shi, Y., Qi, Y., Ma, C., Yuan, R., Wu, D., and Shen, Z.-J. M. (2020). A practical end-to-end inventory management model with deep learning. *Available at SSRN 3737780*.

Quinn, J., Frias-Martinez, V., and Subramanian, L. (2014). Computational sustainability and artificial intelligence in the developing world. *AI Magazine*, 35(3):36–47.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rachev, S., Jašić, T., Stoyanov, S., and Fabozzi, F. J. (2007). Momentum strategies based on reward–risk stock selection criteria. *Journal of Banking & Finance*, 31(8):2325–2346.

Reber, B., Berry, B., and Toms, S. (2005). Predicting mispricing of initial public offerings. *Intelligent Systems in Accounting, Finance & Management: International Journal*, 13(1):41–59.

Reber, B. and Vencappa, D. (2016). Deliberate premarket underpricing and aftermarket mispricing: New insights on ipo pricing. *International Review of Financial Analysis*, 44:18–33.

Ritter, J. R. (1984). The" hot issue" market of 1980. *Journal of Business*, pages 215–240.

Ritter, J. R. (1987). The costs of going public. *Journal of Financial Economics*, 19(2):269–281.

Ritter, J. R. (1991). The long-run performance of initial public offerings. *The journal of finance*, 46(1):3–27.

Robertson, S. J., Golden, B. L., Runger, G. C., and Wasil, E. A. (1998). Neural network models for initial public offerings. *Neurocomputing*, 18(1-3):165–182.

Rock, K. (1986). Why new issues are underpriced. *Journal of financial economics*, 15(1-2):187–212.

Rodriguez, A., ter Horst, E., et al. (2008). Bayesian dynamic density estimation. *Bayesian Analysis*, 3(2):339–365.

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Roosenboom, P. and van der Goot, T. (2005). The effect of ownership and control on market valuation: Evidence from initial public offerings in the netherlands. *International Review of Financial Analysis*, 14(1):43–59.

Rudin, C. and Vahn, G.-Y. (2014). The big data newsvendor: Practical insights from machine learning. *Working Paper*.

Sachs, A.-L. (2015). The data-driven newsvendor with censored demand observations. In *Retail Analytics*, pages 35–56. Springer.

Scarf, H. (1958). A min-max solution of an inventory problem. *Studies in the mathematical theory of inventory and production*.

Scarf, H. (1959). Bayes solutions of the statistical inventory problem. *The Annals of Mathematical Statistics*, 30(2):490–508.

Schmeidler, D. (1989). Subjective probability and expected utility without additivity. *Econometrica: Journal of the Econometric Society*, pages 571–587.

Schmidt, P. and Sickles, R. C. (1984). Production frontiers and panel data. *Journal of Business & Economic Statistics*, 2(4):367–374.

Schwert, G. W. (1989). Why does stock market volatility change over time? *The journal of finance*, 44(5):1115–1153.

Sethi, S. P. and Cheng, F. (1997). Optimality of (s, s) policies in inventory models with markovian demand. *Operations Research*, 45(6):931–939.

Seubert, F., Stein, N., Taigel, F., and Winkelmann, A. (2020). Making the newsvendor smart–order quantity optimization with anns for a bakery chain. *Working Paper*.

Shalit, H. and Yitzhaki, S. (1984). Mean-gini, portfolio theory, and the pricing of risky assets. *The journal of Finance*, 39(5):1449–1468.

Shang, K. H. (2012). Single-stage approximations for optimal policies in serial inventory systems with nonstationary demand. *Manufacturing & Service Operations Management*, 14(3):414–422.

Sharpe, W. F. (1966). Mutual fund performance. *The Journal of business*, 39(1):119–138.

Sheikh, A. Z. and Sun, J. (2012). Regime change: Implications of macroeconomic shifts on asset class and portfolio performance. *The Journal of Investing*, 21(3):36–54.

Silver, E. A., Pyke, D. F., Peterson, R., et al. (1998). *Inventory management and production planning and scheduling*, volume 3. Wiley New York.

Smith Jr, C. W. (1986). Investment banking and the capital acquisition process. *Journal of Financial Economics*, 15(1-2):3–29.

Sodhi, M. S. and Tang, C. S. (2008). The or/ms ecosystem: Strengths, weaknesses, opportunities, and threats. *Operations Research*, 56(2):267–277.

Song, J.-S. and Zipkin, P. (1993). Inventory control in a fluctuating demand environment. *Operations Research*, 41(2):351–370.

Stevenson, R. E. (1980). Likelihood functions for generalized stochastic frontier estimation. *Journal of econometrics*, 13(1):57–66.

Teoh, S. H., Welch, I., and Wong, T. J. (1998). Earnings management and the long-run market performance of initial public offerings. *The Journal of Finance*, 53(6):1935–1974.

Treharne, J. T. and Sox, C. R. (2002). Adaptive inventory control for nonstationary demand and partial information. *Management Science*, 48(5):607–624.

Tsionas, E. G. (2007). Efficiency measurement with the weibull stochastic frontier. *Oxford Bulletin of Economics and Statistics*, 69(5):693–706.

van der Laan, N., Teunter, R. H., Romeijnders, W., and Kilic, O. (2019). The data-driven newsvendor problem: Achieving on-target service levels. Technical report, Working paper, University of Groningen, SOM research school.

Van Parys, B. P., Esfahani, P. M., and Kuhn, D. (2020). From data to decisions: Distributionally robust optimization is optimal. *Management Science*.

Vapnik, V. (1992). Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838.

Wang, L., Fan, H., and Wang, Y. (2018). Sustainability analysis and market demand estimation in the retail industry through a convolutional neural network. *Sustainability*, 10(6):1762.

Welch, I. (1989). Seasoned offerings, imitation costs, and the underpricing of initial public offerings. *The Journal of Finance*, 44(2):421–449.

Whitelaw, R. F. (1994). Time variations and covariations in the expectation and volatility of stock market returns. *The Journal of Finance*, 49(2):515–541.

Xing, F. Z., Cambria, E., and Welsch, R. E. (2018). Intelligent asset allocation via market sentiment views. *ieee ComputatioNal iNtelligeNCe magaziNe*, 13(4):25–34.

Yao, H., Li, Z., and Lai, Y. (2013). Mean–cvar portfolio selection: A nonparametric estimation framework. *Computers & Operations Research*, 40(4):1014–1022.

Yitzhaki, S. (1982). Stochastic dominance, mean variance, and gini's mean difference. *The American Economic Review*, 72(1):178–185.

Young, M. R. (1998). A minimax portfolio selection rule with linear programming solution. *Management science*, 44(5):673–683.

Yung, C., Çolak, G., and Wang, W. (2008). Cycles in the ipo market. *Journal of Financial Economics*, 89(1):192–208.

Zakamulin, V. (2010). The choice of performance measure does influence the evaluation of hedge funds. *SSRN Electronic Journal*, (1403246).

Zhang, Y. and Gao, J. (2017). Assessing the performance of deep learning algorithms for newsvendor problem. In *International Conference on Neural Information Processing*, pages 912–921. Springer.