공학박사 학위논문

Predictive Models for Blockchain, Cryptocurrency, and Derivatives Market

블록체인, 가상화폐, 파생상품 시장을 위한 예측 모형

2018 년 2 월

서울대학교 대학원

산업공학과 산업공학 전공

장 희 수

Predictive Models for Blockchain, Cryptocurrency, and Derivatives Market

블록체인, 가상화폐, 파생상품 시장을 위한 예측 모형

지도교수 이 재 욱

이 논문을 공학박사 학위논문으로 제출함

2017 년 12 월

서울대학교 대학원

산업공학과

장 희 수

장희수의 공학박사 학위논문을 인준함

2018 년 2 월

위 원 장	조 성 준	(인)
부위원장	이 재 욱	(인)
위 원	장 우 진	(인)
위 원	전 치 혁	(인)
위 원	김 남 형	(인)

Abstract

Predictive Models for Blockchain, Cryptocurrency, and Derivatives Market

Huisu Jang
Department of Industrial Engineering
and Naval Architecture
The Graduate School
Seoul National University

This dissertation aims to conduct the empirical analysis for the financial derivative and cryptocurrency market and to develop analytical techniques based on machine learning models suitable for prediction and estimation of each field. In the financial derivative market, a Markov chain Monte Carlo (MCMC) methods employ the candidate probability distribution nearest to the target probability distribution to acquire sample distributed from the posterior density. Choice of the candidate probability distribution affects the practical convergence speed of the MCMC methodology and the fitness of the sample. In this dissertation, we propose a MCMC framework possible to samples from the candidate distribution nearest to the target probability density without the specification of the candidate distribution. We confirm that the jump diffusion models and Bayesian neural networks have the best performance in estimating and predicting given the data of the recent day for the model estimation given S&P index

options in 2012. Especially, the jump diffusion model has a very high performance in terms of domain adaptation between the American option and the European option. This difference is reflected in the fact that the jump diffusion model is based on the common asset of the American option and the European option. Based on this empirical precedent study, we proposed a machine learning model called generative Bayesian neural network (GBNN) to overcome the disadvantages of the machine learning model. GBNN maximizes posterior probability through the GBNN obtains prior information from the GBNN data learned up to the previous day, and learns likelihood probability from actual trading data of learning day. We identify that the GBNN model outperform other benchmark models in terms of model prediction. Bitcoin is a successful cryptocurrency, and it has been extensively studied in fields of economics and computer science. In this dissertation, we analyze the time series of Bitcoin price with a BNN using Blockchain information in addition to macroeconomic variables. We conduct the empirical study that compares the Bayesian neural network with other linear and non-linear benchmark models on modeling and predicting the Bitcoin process. Our empirical studies show that BNN performs well in predicting Bitcoin price time series and explaining the high volatility of the Bitcoin price in Aug. 2017. In addition, we suggested the enhanced GRU model for correlation analysis between cryptocurrency markets. Assuming that the gate value obtained from the GRU model is the parameter of the VAR model, it makes possible to visualize the correlation between various alternative currencies in the cryptocurrency market. As a result, it is confirmed that there is a very significant correlation between the currencies separated from the existing currencies and the existing currencies.

Keywords: financial market analysis, Bayesian neural networks, machine learning, time-series analysis, Markov chain Monte Carlo

Student Number: 2015-30239

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Financial derivative market analysis

Since the seminal work by Black and Scholes (Black & Scholes, 1973) on option pricing in the early 1970's, many alternative option pricing models have emerged to address key stylized facts for option markets, such as volatility smile, fat tail, and volatility clustering. Most of the successful option models are financial jump models (Carr et al., 2003; Kou, 2002; D. B. Madan et al., 1998; Merton, 1976; Nualart et al., 2001) where exact parametric formulas for pricing options are available and can be ready to calibrate to the market data, mostly European option prices, which can be executed only at maturity. American options cannot be priced by the exact closed form since the value of American options includes the right to decide freely to terminate the transaction. Numerous pieces of literature have been studied to circumvent the exact value of American options by employing simulation technology and the assumption of discrete exercise time (Blair et al., 2010; Ederington & Guan, 2002; Fengler, 2006; Kim, 2009; Xu & Taylor, 1995; Benko et al., 2007; Fengler, 2009; BARONE-ADESI & Whaley, 1987; Longstaff & Schwartz, 2001).

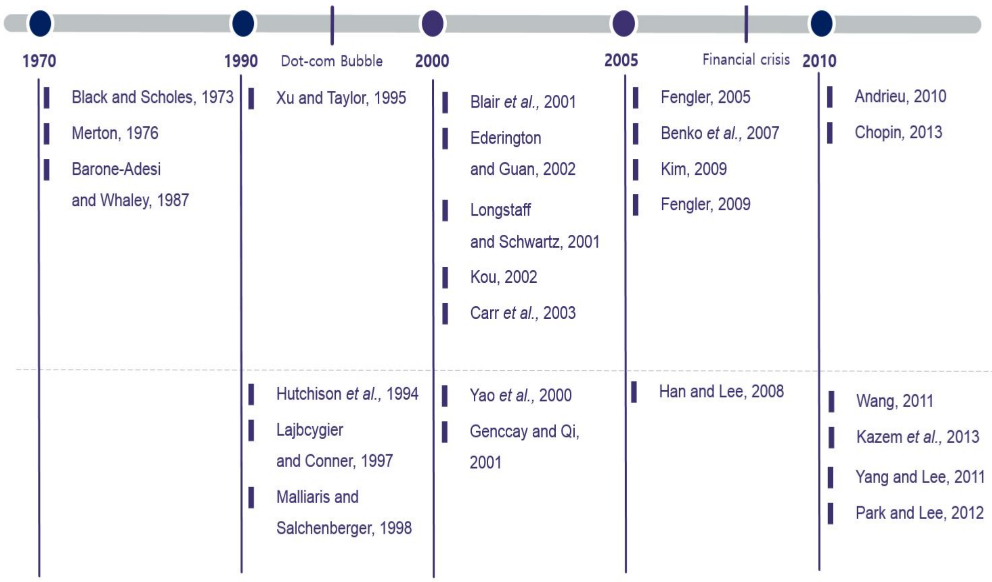


Figure 1.1 Derivative market analysis based on econometric and AI models

The tremendous increase in computing power and data storage during the last decade has resulted in the rapid development of machine learning and data mining with diverse applications in economics, finance, science, engineering, and technology. In the finance area, machine learning models have elicited considerable attention from many researchers because of their predictive power. Yao and Tan (2000b) demonstrated that Nikkei 225 index future options in 1995 were better predicted by neural networks using the back-propagation algorithm than the traditional Black-Scholes models. Gençay and Qi (2001a) showed that generalization for pricing and hedging derivatives can be improved by the Bayesian regularization techniques and verified empirically for S&P 500 index daily call options from January 1988 to December 1993. Wang (2011a) reported that support vector regression (SVR) improved the forecast accuracy for the daily currency market data of AUD/USD, EUR/USD, USD/JPN, and GBP/USD options from January to July in 2009. Kazem et al. (2013) presented support vector regression methods optimized by chaotic firefly algorithm outperforms several methods of SVR for NASDAQ quotes, Intel (from 9/12/2007 to 11/11/2010), National Bank shares (from 6/27/2008 to 8/29/2011) and Microsoft (from 9/12/2007 to 11/11/2011) daily closed stock prices. Xiong et al. (2014) tuned the parameters of multi-output support vector regression using firefly algorithm and compared the proposed SVR methods with other existing methods for forecasting the market indexes, S&P 500, Nikkei 225, and FTSE 100 indexes. Figure 1.1 shows the derivative market analysis based on econometric and AI models according to time line.

Although a considerable number of studies have been conducted to elucidate financial option markets by applying either econometric financial models or machine learning models, there are few studies considering two types of the models jointly. In this dissertation, an intensive empirical study is conducted to compare econometric models with state-of-the-art machine learning models especially concentrating on the American option pricing analysis relatively having the limitation in only use traditional econometric models. In addition to empirical studies, we suggested the proposed MCMC methodologies to parameter calibration of time series models to acquire suitable parameters. We propose a generative Bayesian machine learning model to improve the options market predictability.

1.2 Cryptocurrency market analysis

In this dissertation, the cryptocurrency market refers to the decentralized cryptocurrency market produced by the entire cryptocurrency system collectively, based on the underlying technical system created by Satoshi Nakamoto. The decentralized cryptocurrency, without the intrinsic value, is valued by shared ledger among participants unlike fiat currencies, which are valued by the central banking and economic system like the Federal Reserve System (FRB) capable of controlling the money supply. It is inevitable to consider the mechanism of the 'shared ledger' techniques, 'Blockchain', for the cryptocurrency analysis, since the Blockchain techniques are directly involved in the supply and demand of the cryptocurrency. Currently, there is very little research dedicated to the first currency market rooted in the technology other than small amounts of studies about the Bitcoin by employing existing econometric technologies.

Numerous studies have been conducted recently on modeling the time series of Bitcoin prices based on existing econometric models under the assumption the Bitcoin is regarded as a general market variable. Generalized Autoregressive Conditional Heteroskedasticity (GARCH) volatility analysis is performed to explore the time series of Bitcoin price (Dyhrberg, 2016a),(Katsiampa, 2017). Various studies on statistical or economical properties and characterizations of Bitcoin prices refer to its capabilities as a financial asset; these research focus on statistical properties (Bariviera et al., 2017; Chu et al., 2015), inefficiency of Bitcoin according to efficient market hypothesis (Urquhart, 2016; Nadarajah & Chu, 2017), hedging capability (Dyhrberg, 2016b; Bouri et al., 2017), speculative bubbles in Bitcoin (Cheah & Fry, 2015), the relationship between Bitcoin and search information, such as Google Trends and Wikipedia (Kristoufek, 2013), and wavelet analysis of Bitcoin (Kristoufek, 2015).

Relatively few studies have thus far been conducted on estimation or prediction of Bitcoin prices. Ciaian et al. (2016) evaluates Bitcoin price formation based on a linear model by considering related information that is categorized into several factors of market forces, attractiveness for investors, and global macro-financial factors. They assume that the first and second factors mentioned above significantly influence Bitcoin prices but with variation over time. The same researchers limit the number of regressors to facilitate linear model analysis. McNally (2016) predicts the Bitcoin pricing process using machine learning techniques, such as recurrent neural networks (RNNs) and long short-term memory (LSTM), and compare results with those obtained using autoregressive integrated moving average (ARIMA) models. A machine trained only with Bitcoin price index and transformed prices exhibits poor predictive performance. I. Madan et al. (2015) compares the accuracy of predicting Bitcoin price through binomial logistic regression, support vector machine, and random forest.

There are few practical and systematic empirical studies on the analysis of cryptocurrency markets. We conduct practical analysis on modeling and predicting of the Bitcoin process by employing a Bayesian neural network (BNN), which can naturally deal with increasing number of relevant features in the evaluation based on Blockchain information. We also try to account for the remarkable coupling of other cryptocurrencies with the Bitcoin by employing the enhanced GRU framework.

1.3 Aims of the Dissertation

This dissertation aims to investigate the intensive analysis for financial derivative market and cryptocurrency market and to develop analytic technologies based on AI strategies suitable for prediction and estimation of each field. We evaluate the suitability of the econometric models and the data-driven machine learning models for each specific market analysis by bringing major two criteria of model validity and model predictability. Model validity considers the in-sample error value and parameter calibration results, and model predictability will verify that the model forecasts reasonable prices for the out-of-sample. In addition to the intensive empirical study to compare econometric models with state-of-the-art machine learning models, we propose the machine learning models suitable for market analysis based on the advantages and limitations of each econometric and machine learning models acquired from the precedence studies. We conducted parallel studies for the each index options market and the cryptocurrency market.

In Chapter 2, econometric models and statistical machine learning models used in this dissertation are summarized at first.

In Chapter 3, we propose the generative Markov chain Monte Carlo (MCMC) framework for parameter calibration of state space models. It achieves more stable parameter estimation of considered econometric models. Then, an intensive empirical study is conducted to compare two methods in terms of model estimation, prediction, and domain adaptation using S&P 100 American/European put options. Results indicated that econometric jump models demonstrate better prediction performance than the best-performing machine learning models, and the estimation results of the former are similar to those of the latter. The former also exhibited significantly better domain adaptation performance than the latter regardless of domain adaptation techniques in machine learning. Lastly, we propose a generative Bayesian neural network model that incorporates a prior reflecting a risk-neutral pricing structure to be consistent with the extreme option prices. Proposed model can acquire the information under the extreme region where there are a few observations in real data by considering artificial prior information from the econometric model. Chapter 4 included a real data application to compare the proposed model with other state-of-the-art methods in terms of model estimation and prediction using S&P 100 American put options data from 2003 to 2012.

In Chapter 4, we investigate the cryptocurrency market analysis based on the real data empirical study of Bitcoin. We conduct the empirical study that compares the Bayesian neural network with other linear and non-linear benchmark models on modeling and predicting the Bitcoin process. Our empirical studies show that BNN performs well in predicting Bitcoin price time series and explaining the high volatility of the recent Bitcoin price. We also the enhanced GRU framework based on the Vector Autoregressive (VAR) model to reveal the relationships of several cryptocurrencies and shows the correlation between several cryptocurrencies over time by conducting real data experiments based on the eight types of cryptocurrency data. There has been limited research on the machine learning framework of interpretable data based. This dissertation focuses on visualizing and interpreting meaning from the data by developing a machine learning based model easy to interpret. This dissertation can contribute to the primary data analysis for the cryptocurrency market.

Finally, we discuss the contributions and future works of this dissertation in Chapter 5.

1.4 Outline of the Dissertation

To achieve the aims of the dissertation, the rest of this dissertation is organized as the following Table 1.1. Table 1.1 summarizes analysis market fields and analysis scopes in the entire discussion.

Table 1.1 Analysis market fields and scopes in the dissertation

Market fields	**Analysis scope**	**Included chapter**
Options market	Empirical comparison study	**Chapter 3.2**
	Model validity	**Chapter 3.3**
	Model predictability	**Chapter 3.4**
Cryptocurrency market	Empirical comparison study	**Chapter 4.3**
	Model validity	**Chapter 4.4**

Chapter 2

Literature Review

2.1 Review of Financial Econometric Models

2.1.1 Time series models

Financial jump models divide into finite-activity jump-diffusion processes and infinite-activity exponential Lévy processes, respectively. The former includes the Merton model with finitely Gaussian jumps (Merton, 1976) and the Kou model with double exponential jumps (Kou, 2002). The latter includes the Variance Gamma (VG) (D. B. Madan & Seneta, 1990; D. B. Madan et al., 1998), the (Generalized) Hyperbolic model or the Normal Inverse Gaussian(NIG) (Eberlein et al., 1995; Barndorff-Nielsen, 1997), the CGMY (named after Carr, German, Madan and Yor) (Carr et al., 2003), and Meixner model (Nualart et al., 2001).

A representative jump model we consider in this paper is the CGMY model (also called truncated Lévy flights) which is an infinite activity exponential-Lévy process (Carr et al., 2003) given by the following risk-neutral stock price process

$$S_t = S_0 exp((r - q)t + X_t(\nu) + \omega t)$$

where r and q represent the constant continuously compounded interest rate and dividend yield respectively, $X_t(\nu)_{t\geq 0}$ is a Lévy process. Lévy measure ν, and ω is instituted to guarantee the martingale property for the price process. Lévy measure of the CGMY model takes the form of

$$\nu(x) = \frac{c}{(-x)^{1+Y}} e^{\lambda_- x} 1_{x<0} + \frac{c}{x^{1+Y}} e^{-\lambda_+ x} 1_{x>0}$$

It is of finite variation if $0 \leq Y < 1$ and of infinite variation if $Y \geq 1$.

There are 4 parameters $\theta = (c, Y, \lambda_-, \lambda_+)$: c determines the overall and relative frequency of jumps, λ_-, λ_+ represent the tail behavior of the Lévy measure, Y shows the local behavior of the process (how the price evolves between big jumps).

The characteristic function of the model is sufficient to apply numerical approximation for pricing options. The characteristic function $\Phi_s(z)$ of $s_t = \ln(S_t/S_0)$ is represented by

$$\Phi_s(z) = \mathbb{E}[e^{izs_t}] = e^{t(iz(r-q+\omega)+tc\Gamma(-Y)((\lambda_+-iz)^Y-\lambda_+^Y+(\lambda_-+iz)^Y-\lambda_-^Y)} \tag{2.1}$$

and

$$\omega = c\Gamma(-Y)((\lambda_+ - 1)^Y - \lambda_+^Y + (\lambda_- + 1)^Y - \lambda_-^Y)$$

where $0 < Y < 1$ or $Y > 1$ and $\Gamma(-Y)$ means a gamma function value of $-Y$.

The Kou model (Kou, 2002) is a econometric jump model that includes a jump term with known distribution of jump sizes that describes abnormal rare market events. The dynamics of stock price is given by the following stochastic differential equation(SDE):

$$\frac{dS_t}{S_t} = (\gamma - q)dt + \sigma dW_t + d(\sum_{i=1}^{N(t)} (V_i - 1))$$

where γ and q represent the constant continuously compounded interest rate and dividend yield respectively, W_t is a Brownian motion, $N(t)$ is a Poisson processes with parameter λ, and V_t is a sequence of i.i.d. non-negative random variables. The distribution $Y_t = \ln(V_t)$ of jump sizes is an asymmetric exponential as follows:

$$p_Y(x) = p\lambda_+ e^{-\lambda_+ x} 1_{x>0} + (1-p)\lambda_- e^{\lambda_- x} 1_{x<0}$$

where the tail behavior of positive and negative jump sizes distribution is considered by $\lambda_\pm > 0$ and $p \in [0,1]$ represents the probability of an upward jump.

There are 5 parameters $\theta = (\lambda, \lambda_+, \lambda_-, p, \sigma)$: λ, jump intensity, λ_+, λ_-, p, parameters of each jump size distribution, and σ, diffusion volatility. The characteristic function $\Phi_s(z)$ of $s_t = \ln(S_t/S_0)$ is given by

$$\Phi_s(z) = \mathbb{E}[e^{izs_t}] = e^{t(iz(r-q+\omega_0)-\frac{1}{2}z^2\sigma^2+\lambda(p\frac{\lambda_+}{\lambda_+-iz}+(1-p)\frac{\lambda_-}{\lambda_--iz}-1))} \tag{2.2}$$

and

$$\omega_0 = -\frac{1}{2}\sigma^2 + \lambda\left(1 - p\frac{\lambda_+}{\lambda_+ - 1} - (1-p)\frac{\lambda_-}{\lambda_- + 1}\right)$$

In this dissertation, the parameters of the CGMY model are calibrated from minimizing the mean squared error between true prices and estimated prices. Pricing the American option prices can be achieved by applying the Fourier Cosine method or solving the linear complementarity problem (LCP). (Kwon & Lee, 2011) suggested the implicit method coupled with the operator splitting method to preserve the second order accuracy in the time and spatial variables. The numerical method called *the implicit method with three time level* has the advantage that it avoids the iteration we need to solve the dense linear system at each time step. Then the prices of the American options can be evaluated with the computational complexity of $O(MN \log_2 M)$ operations where M is the number of spatial steps and N is the number of time steps. The Fourier Cosine method is one of the efficient pricing methods for the European options (Fang & Oosterlee, 2008). They also propose the Fourier-Cosine series method for pricing early-exercise and discrete barrier options (Fang & Oosterlee, 2009). The computational complexity is O((M-1)NlogN) with a number of the series expansion, N, the number of monitoring dates, M. They dose not present the parameter estimation results of the proposed method in the literature.

In financial time series analysis, time-varying volatility is often considered to mitigate the drawbacks of the deterministic volatility by employing the stochastic volatility (SV) models. We consider the representative two SV models: the Heston model (Heston, 1993) and the generalized autoregressive conditional heteroskedasticity (GARCH) model (Bollerslev, 1986). The Heston model assumes that the asset price, S_t, and the instantaneous variance, v_t at time t are determined by the following process:

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_t^1 \\ dv_t &= \kappa(\theta - v_t)dt + \eta\sqrt{v_t} dW_t^2 \end{aligned}$$

where dW_t^1 and dW_t^2 are Wiener processes with correlation ρ and the parameter set, $(\mu, \kappa, \theta, \eta)$, describes the rate of return, the reverting rate of variance, the long run average price variance, and the volatility of the volatility.

We consider the standard GARCH(1,1) model, which assumes that the randomness of the variance process varies with the variance. The standard

GARCH(1,1) model is similar to the Heston model except that the square root value is removed from the process of variances. The processes of GARCH(1,1) model are as follows:

$$
\begin{aligned}
dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_t^1 \\
dv_t &= \kappa(\theta - v_t)dt + \eta v_t dW_t^2
\end{aligned}
$$

where dW_t^1 and dW_t^2 are independent Wiener processes and the parameter set, $(\mu, \kappa, \theta, \eta)$, describes the rate of return, the reverting rate of variance, the long run average price variance, and the volatility of the volatility.

Calibration issue of stochastic volatility models is challenging since the volatility process of an asset return is not directly observed. (Bakshi et al., 1997) used implied volatilities and cross-sectional information in option prices with different maturities and strike prices to estimate the asset return and stochastic volatility process. We followed the two-step calibration procedure applied in (AitSahlia et al., 2010; Zhang & Shu, 2003): first, structural parameters of the underlying asset is approximated by the indirect inference methods from (Gourieroux et al., 1993), and the rest parameters for option pricing are estimated based on least-squares by using market European options. After the calibration, we employ the least-squares Monte-Carlo (LSM) method to price American options to avoid the stability problem of partial differential equations.

In what follows we shall provide a selective overview of some popular classical American option pricing methods in financial studies.

2.1.2 Option pricing methods

Regression approaches introduced by (Longstaff & Schwartz, 2001) have applied to estimate the continuation values of American or other exotic options(Clément et al., 2002; Tsitsiklis & Van Roy, 2001). The approach assumes that the options can be exercised at m discrete time set $0 \leq t_1 \leq t_2 \leq \ldots \leq t_m = T$, where T is the expiration date. The method starts with N random paths $\{S^n_{t_i}(\omega_n)|i = 1, \cdots, m\}$ according to Markov chain for $1 \leq n \leq N$ and determines option values by rolling-back on these paths. The value of American option, $V(t_i)$, at each time step t_i is then defined recursively by the following forms,

$$V(t_i) = \max[X(t_i), \mathbb{E}(V(t_{i+1}|\mathcal{F}(t_i))], \quad i = 1, \ldots, m-1. \tag{2.3}$$

where $X(t_i)$ means that the immediate payoff at time t_i, and $\mathcal{F}(t_i)$ is the σ-algebra until time t_i. The conditional expectation of continuation value for each path, which is the value of holding rather than exercising, is given by $C(\omega, t_i)$ and is estimated by the fitted values of the following regression:

$$C(\omega, t_i) = \sum_{j=0} \beta_j B_j(S_{t_i}(\omega)), \tag{2.4}$$

where β_j is the coefficients of the regression function, and $B_j(\cdot)$ is the basis functions with the underlying asset value S_{t_i} at time t_i. There are many possible choices of basis functions such as Laguerre, Hermite, Legrendre, and Jacobi polynomials. In this experiment, we adopt the Laguerre poynomials of the second term:

$$\begin{aligned}
B_0(S) &= \exp(-S/2) \\
B_1(S) &= \exp(-S/2)(1-S) \\
B_2(S) &= \exp(-S/2)(1-2S+S^2/2) \\
&\vdots \\
B_n(S) &= \exp(-S/2)\frac{e^S}{n!}\frac{d^n}{dS^n}(S^n e^{-S})
\end{aligned} \tag{2.5}$$

The ordinary least-square regression calculates the estimates of regression coefficients at each time t_i. The price of the American option is then calculated by averaging $F(\omega, 0)$ over all ω paths.

Barone-Adesi and Whaley model is one of the most widely used analytic approximation method for pricing American options (BARONE-ADESI & Whaley, 1987). In Barone-Adesi Whaley models, American option values are expressed as a sum of European option values and early exercise premium, and can be obtained by computing critical values where American and European put option values are indifferent. The model assumes that the underlying process follows a geometric Brownian motion with constant volatility σ as with (Black & Scholes, 1973), and that the risk free interest rate, r, and the cost of carrying the underlying, b, which is equal to the difference with risk free rate and the dividend yield, d (i.e., $b = r - d$) are all constants.

We define the early exercise premium with expiration date T, and strike price K as

$$v(S, K) = P_{American}(S, T) - P_{European}(S, t) = h(T-t)f(S, K)$$

by choosing $h(T-t) = 1 - e^{-r(T-t)}$. Then the approximate value of an American put option is

$$P_{American}(S,T) = \begin{cases} P_{European}(S,T) + A_1 \left(\frac{S}{S^*}\right)^{q_1}, & \text{when} \quad S > S^* \\ X - S, & \text{when} \quad S \leq S^* \end{cases} \tag{2.6}$$

where S^* is the critical underlying value below which the option should be exercised and can be calculated numerically by solving following equation

$$X - S^* = P_{European}(S^*, t) - \frac{S^*}{q_1}[1 - e^{(b-r)(T-t)} N[-d_1(S^*)]]$$

The other variables are given by

$$\begin{aligned} q_1 &= \frac{1}{2}\left[-(\frac{2b}{\sigma^2} - 1) - \sqrt{(\frac{2b}{\sigma^2} - 1)^2 + \frac{8r}{h\sigma^2}}\right] \\ A_1 &= -\left(\frac{S^*}{q_1}\right)(1 - e^{(b-r)(T-t)} N[-d_1(S^*)]) \\ d_1(S) &= \frac{\ln(S/K) + (b + \sigma^2)(T-t)}{\sigma\sqrt{T-t}} \end{aligned}$$

Note that $A_1 > 0$ since $q_1 < 0$, $S^* > 0$, and $N[-d_1(S^*)] < e^{-b(T-t)}$.

Option pricing using implied and local volatility have shown superior predictive power (Blair et al., 2010; Ederington & Guan, 2002; Fengler, 2006; Kim, 2009; Xu & Taylor, 1995).

The ad hoc Black-Scholes method employs the implied volatilities smoothed across strike prices and time to maturity that are plugged back into the Black-Scholes formula to cope with the volatility smile effects. It is one of the most widely used option evaluation techniques among practitioners for its consistently impressive empirical performance of option evaluation. Brandt and Wu (2002) demonstrated that the ad hoc Black-Scholes outperformed the deterministic volatility function models in evaluating FTSE 100 index options. Heston and Nandi (2000) compared the ad hoc approaches with GARCH models. Christoffersen and Jacobs (2004) reported that the ad hoc methods with daily parameter update is better than Heston (1993)'s model for in sample and out of sample.

The ad hoc Black-Scholes methods approximate the surface of implied volatilities by a regression against a polynomial function of the strike prices and time to maturities of the options,

$$\hat{\sigma}_{IV} = \beta_0 + \beta_1 K + \beta_2 T + \beta_3 KT + \epsilon \quad \epsilon \sim N(0,1) \tag{2.7}$$

where K is the strike price and T is the time to maturity. Approximated implied volatilities $\hat{\sigma}_{IV}$ have no sensible interpretation, but when plugged back into the Black-Scholes formula it gives the option value, that is,

$$BS(K, T, S, r, \hat{\sigma}_{IV}(K, T)) \tag{2.8}$$

where the strike price, K, the expiration date, T, the underlying value, S, the risk-free rate, r, and the approximated volatility, $\hat{\sigma}_{IV}(K, T)$ from the regression. It is different from the true own implied volatility for each option values, nevertheless, is believed to capture some implication of a free parameter across concurrent options. Berkowitz et al. (2010) showed that asymptotic argument is valid for American options if the volatilities are estimated from American options.

Local fit with more flexible nonparametric smoothing methods have recently been employed for functional flexiblility (Benko et al., 2007; Fengler, 2006, 2009). Unlike the case of popular parametric stochastic volatility models that always satisfy the no-arbitrage conditions, implied volatility surfaces requires some constraints to be arbitrage-free.

Local volatility is defined as the function of the underlying asset price and at any given time. Additionally, we assume that the underlying asset is consistent with the following stochastic process:

$$dS_t = (r-\delta)S_t dt + \sigma(S_t, t)dW_t^{\mathbb{Q}} \tag{2.9}$$

Dupire (1997) and Gatheral (2011) formulated the local volatility function by giving the total implied variance. The total implied variance(TIV) is defined as $v(y,T) = \sigma_{IV}^2(y,T)T$ with the implied volatility, $\sigma_{IV}(\cdot,\cdot)$, the log-forward moneyness, $y = ln(K/F_T)$, F_T is the forward price, and , the time to maturity, T. Then the local volatility surfaces satisfies the following equation:

$$\sigma^2(T, F_T S_0 e^y) = \frac{\frac{\partial v}{\partial T}(T,y)}{1 - \frac{y}{v}\frac{\partial v}{\partial y}(T,y) - \frac{1}{4}(\frac{1}{4} + \frac{1}{v} - \frac{y^2}{v})\frac{\partial v}{\partial y}^2(T,y) + \frac{1}{2}\frac{\partial^2 v}{\partial y^2}(T,y)} \tag{2.10}$$

There are several nonparametric smoothing methods to estimate the local volatility surface by employing the polynomials, piecewise polynomials, and functional forms (Dumas et al., 1998; Benko et al., 2007; Hastie et al., 2009; Fengler, 2006). We utilized a bivariate local quadratic kernel smoothing easy to acquire the derivatives of the total implied variances, $v(y,T)$(Benko et al., 2007; Fengler, 2006). Then it can be estimated by optimizing the following minimization problem,

$$\begin{aligned} \min_{\beta} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n}(\beta_0 + \beta_1(y_i - y) + \beta_2(T_j - T) + \beta_3(y_i - y)(t_j - t) \\ & + \beta_4(y_i - y)^2 - v(y_i, T_{,j}))^2 K(y_i - y, T_j - T) \end{aligned} \tag{2.11}$$

where $\{(y_i, T_j, v(y_i, T_j)); \quad i = 1, \cdots, m, \ j = 1, \cdots, n\}$ is the given data set, $\beta = (\beta_0, \cdots, \beta_4)^T$ is the coefficients of estimates, and $K(y,T) = K_{i_1}(y)K_{i_2}(T) = \exp(-y^2/(2i_1^2))/(i_1\sqrt{2\pi})\exp(-T^2/(2i_2^2))/(i_2\sqrt{2\pi})$ is the bivariate kernel functions. To calculate the local volatility, the derivatives of the total implied variances can be easily obtained from the estimation results:

$$\hat{v}(y,T) = \hat{\beta}_0(y,T), \quad \frac{\partial \hat{v}(y,T)}{\partial y} = \hat{\beta}_1(y,T),$$
$$\frac{\partial \hat{v}(y,T)}{\partial T} = \hat{\beta}_2(y,T), \quad \frac{\partial^2 \hat{v}(y,T)}{\partial y \partial T} = \hat{\beta}_3(y,T), \quad \frac{\partial^2 \hat{v}(y,T)}{\partial y^2} = 2\hat{\beta}_4(y,T).$$

MC simulations are performed under the above local volatility model, and the fast Fourier transform with the Heston's model to price the option prices. In current experiment, we put the implied volatilities of OEX put options from the Optionmetrics database by the Wharton Research Data Services.

2.2 Review of Statistical Machine Learning Models

In this section, state-of-the-art machine learning models, such as artificial NNs, support vector machines, and GPs, are briefly reviewed as below.

2.2.1 Artificial neural networks

An artificial neural network (ANN) popularized after mid-1980s and now in the 2010s with another name of deep learning has been successful in many applications such as image recognition, speech recognition, natural language processing, and financial time series (Murphy, 2012). The structure of an ANN which mimics human brain structure consists of several connected layers where each layer is the aggregate of neurons which are connected to each other. Layers except for the input and output layer are referred to the hidden layer where each hidden or output layer represents mathematically a nonlinear function of the linear combination of the neuron node values that are delivered forward from input nodes or hidden nodes. The employed nonlinear function is referred as activation function such as hyperbolic tangent or logistic function. See Figure 2.1 below. The function form of the trained ANN model in this network diagram

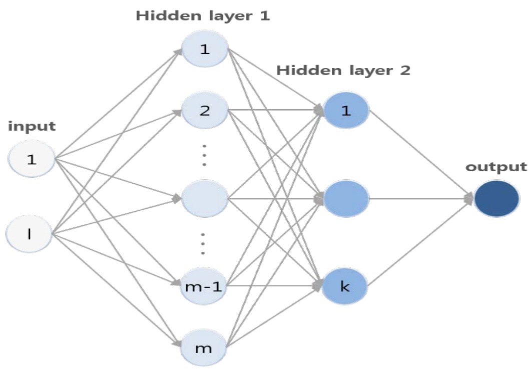


Figure 2.1 Deep neural network with two-hidden-layers.

is represented as,

$$ff(\mathbf{x}, \mathbf{w}) = \sigma(\sum_{j=1}^{k} w_j^{(3)} h_2(\sum_{i=1}^{m} w_{ji}^{(2)} h_1(\sum_{t=1}^{l} w_{it}^{(1)} x_t + w_{i0}^{(1)}) + w_{j0}^{(2)}) + w_0^{(3)}) \quad (2.12)$$

where $w_{ij}^{(k)}$ in the set of weight vectors $\mathbf{w}$ means a weight between the j-th variable in the $(k-1)$-th hidden layer and the i-th variable in the k-th hidden layer, the 0-th hidden layer refer to the input layer, the last k-th hidden layer is the output layer, $h_i(\cdot)$ is the i-th hidden activation function, $\sigma(\cdot)$ is the sigmoid function of the output layer for the regression, and each x_t means the t-th variable in the input vector $\mathbf{x}$.

Backpropagation algorithm is the most popular gradient descent method to train the model by changing the weights of neural networks to reduce the chosen error function between the model predicted value and the true output, generally mean squared error. The gradient value use in this algorithm takes the form of a product of each partial derivative element of which the total value is rapidly shrinking to zero when the number of partial derivative terms is increasing, thereby causing frequently a vanishing gradient problem in training multilayer neural networks or deep neural networks.

Bayesian neural network is another popular class of neural networks proposed to mitigate the over-fitting problem (Burden & Winkler, 2009; MacKay, 1992) by adding a Bayesian regularization term to objective function as follows:

$$F = \beta \sum ||y - f(\mathbf{x}, \mathbf{w})||^2 + \alpha \sum ||\mathbf{w}||^2 \tag{2.13}$$

where F is the objective function, y is the output data, and $\mathbf{w}$ is the weights of the network which are random variables with a density function given by

$$P(\mathbf{w}|D, f) = \frac{P(D|\mathbf{w}, f)P(\mathbf{w}|f)}{P(D|f)} \tag{2.14}$$

where D is the training data for neural networks. Levenburg-Marquardt algorithm are used to find the weights of networks to achieve minimization of the objective function (Foresee & Hagan, 1997) and determines regularization parameters α and β by approximating the Hessian matrix of objective function at the minimum point. This technique increases the robustness of the model by mitigating the local minimum problem.

2.2.2 Bayesian neural networks

Bayesian neural networks (BNN) is a transformed Multilayer perceptron (MLP) which is a general term for ANNs in the fields of machine learning. The networks have been successful in many application such as image recognition, pattern recognition, natural language processing, and financial time series (Murphy, 2012). It becomes known that much effective to represent the complex time series than the conventional linear models, i.e. autoregressive and moving average, etc. The structure of a BNN is constructed with a number of processing units classified into three categories: an input layer, an output layer, and one or more hidden layers.

Specifically, neural networks containing more than one hidden layers can solve the exclusive OR (XOR) problem, which cannot be solved by a single layer perceptron (Minsky & Papert, 1969). Different from a single layer perceptron, which can only be linearly separated, they solve XOR problems by introducing backpropagation algorithms and hidden layers. The hidden layer mapping the original data to a new space transforms data that cannot be linearly separated into linearly separable data.

Weights of a BNN must be learned between the input-hidden layer and hidden-output layer. Backpropagation refers to the process in which weights of hidden layers are adjusted by the error of hidden layers propagated by the error of the output layer. An optimization method called delta rule is used to minimize the difference between a target value and output value when deriving backpropagation algorithm. In general, BNNs minimize the sum of the following errors, E_B, using backpropagation algorithm and delta rule.

$$E_B = \frac{\alpha}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} (t_{nk} - o_{nk})^2 + \frac{\beta}{2} \precapprox_B^T \precapprox_B \tag{2.15}$$

where E_B is the sum of the errors, N is the number of the training variables, K is the size of the output layer, t_{nk} is the k-th variable of the n-th target vector, o_{nk} is the k-th output variable of the n-th training vector, α and β are the hyper-parameter, and $\precapprox_B$ is the weights vector of the Bayesian neural network.

A BNN is a non-linear version of ridge regression, which is largely based on the Bayesian theory for neural networks. Unlike conventional neural networks that maximize marginal likelihood, BNN is a machine maximizing the value of posterior through an application of the Bayes' theory. The elements added to the error term cause the machine to learn by selecting a weight with high importance even when the number of total weights is reduced rather than distributed to a large number of weights.

2.2.3 Support vector regression

Support vector machine (SVM) is a state of the art kernel machine learning method and is successfully applied to nonlinear classification, regression, and clustering problems (Vapnik, 2013). Given a set of observations $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, support vector regression (SVR) model aims at finding a regression function value $f(\mathbf{x}_i)$ that has smaller deviation than predetermined ϵ from the targets y_i. Specifically, SVR begins with nonlinear functions f of the form

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}\rangle + \mathbf{b} \tag{2.16}$$

where $\langle , \rangle$ denotes the inner dot product and Φ is a nonlinear map from an input space into a feature space. Then it try to find the flatness which can be achieved by minimizing the weight norm subject to the deviation is at most ϵ as follows.

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\
\text{subject to} \quad & y_i - \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \leq \epsilon + \xi_i \\
& \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b - y_i \leq \epsilon + \xi_i^* \\
& \xi_i, \xi_i^* \geq 0
\end{aligned} \tag{2.17}$$

where the constant $C > 0$ is concerned with the trade off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated. In this formulation, slack variables ξ_i, ξ_i^* allow some error bigger than ϵ to deal with infeasible constraints of the problem. This primal optimization problem can be efficiently solved using the so-called kernel tricks by solving its dual problem as

$$\begin{aligned}
\text{maximize} \quad & -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i*)(\alpha_j - \alpha_j*)k(\mathbf{x}_i, \mathbf{x}_j) \\
& -\epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i*) \\
\text{subject to} \quad & \sum_{i=1}^{l}(\alpha_i - \alpha_i*) = 0 \\
& \alpha_i, \alpha_i* \in [0, C]
\end{aligned} \tag{2.18}$$

where a kernel defined by $k(\mathbf{x}, \mathbf{x}') := \langle(\Phi(\mathbf{x}), \Phi(\mathbf{x}')\rangle$ is used instead of $\Phi(.)$ explicitly. In this paper, we adopt the most popular RBF kernel which is defined by

$$k(\mathbf{x_i}, \mathbf{x_j}) = exp(-\gamma||\mathbf{x_i} - \mathbf{x_j}||^2), \quad \gamma > 0 \tag{2.19}$$

(See more details in (Vapnik, 2013).)

2.2.4 Gaussian process

Gaussian process (GP) is a state of the art Bayesian kernel regression model and is defined by a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen & Williams, 2006). A GP $f(\mathbf{x})$ for an observed input $\mathbf{x}$ is regarded as function value vector sampled from a multivariate Gaussian distribution over the space of functions. It has a mean function $\mathbf{m}(\mathbf{x})$ and a covariance function $\mathbf{k}(\mathbf{x}, \mathbf{x}')$ given by

$$\begin{aligned} \mathbf{m}(\mathbf{x}) &= E[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= E[(f(\mathbf{x}) - \mathbf{m}(\mathbf{x}))(f(\mathbf{x}') - \mathbf{m}(\mathbf{x}'))] \end{aligned} \tag{2.20}$$

Given a set of observations $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the joint distribution of $\mathbf{y} = (y_1, ..., y_N)^T$, the covariance of $\mathbf{y}$ is represented by

$$\text{cov}(\mathbf{y}) = \mathbf{K} + \sigma^2\mathbf{I} \tag{2.21}$$

where $\mathbf{K}$ is an $N \times N$ covariance matrix with its ij-th component $k(\mathbf{x}_i, \mathbf{x}_j)$. Gaussian process has the following joint distribution of $\mathbf{y}$ and $f* = f(\mathbf{x}^*)$ for a new input $\mathbf{x}^*$;

$$\begin{pmatrix} \mathbf{y} \\ f* \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k}_*^T \\ \mathbf{k}_* & k_{**} \end{bmatrix}), \tag{2.22}$$

where $\mathbf{k}_{**} = k(\mathbf{x}^*, \mathbf{x}^*), \mathbf{k}_* = (k(\mathbf{x_1}, \mathbf{x}^*), ..., k(\mathbf{x_N}, \mathbf{x}^*))^T$. The conditional distribution of $f*$ given $\mathcal{D}$ then follows the Gaussian distribution as

$$f * | \mathcal{D} \sim \mathcal{N}(\mathbf{k}_*^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, k_{**} - \mathbf{k}_8^T(\mathbf{K} = \sigma^2\mathbf{I})^{-1}\mathbf{k}_*) \tag{2.23}$$

Now, we can check that the posterior variance $k^*(\mathbf{x}, \mathbf{x}')$ is independent of $\mathbf{y}$. There are several widely used covariance functions $k(\mathbf{x}, \mathbf{x}')$ whose choice is quite dependent on the characteristics of the problems. We chose the "Matérn class" as covariance functions in our experiments as (G.-S. Han & Lee, 2008), which is given by

$$k(\mathbf{x}, \mathbf{x}') = \frac{1}{\Gamma(\nu)2^{\nu-1}}[\frac{\sqrt{2\nu}}{l}||\mathbf{x} - \mathbf{x}'||]^\nu K_\nu(\frac{\sqrt{2\nu}}{l}||\mathbf{x} - \mathbf{x}'||) \tag{2.24}$$

where ν and l are positive parameters, and K_ν is a modified Bessel function with ν which controls the degree of smoothness. The complete specification of the GP can be achieved by maximizing the marginal log likelihood over hyperparameter $\theta = (\nu, l)$:

$$\ln \mathbb{P}(\mathbf{y}|\mathcal{D}, \theta) = -\frac{1}{2}\mathbf{y}^T(\mathbf{K} = \sigma^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}ln|\mathbf{K} + \sigma^2\mathbf{I}| - \frac{N}{2}log2\pi \tag{2.25}$$

(See (Rasmussen & Williams, 2006) for more details on the GP.)

Chapter 3

Predictive Models for the Derivatives Market

3.1 Chapter Overview

Machine learning models, which are equipped with outstanding predictability, have applied to financial forecasting, especially in financial derivatives market. Most machine learning methods forecasted the prices of financial derivatives with the expectation that the process of underlying assets will be represented implicitly as a learning function of input variables without the explicit form of return processes. Successful machine learning models for predicting financial derivatives include artificial neural networks (NNs) (Hutchinson et al., 1994; Lajbcygier & Connor, 1997; Malliaris & Salchenberger, 1996; Yao & Tan, 2000a; Gençay & Qi, 2001b), support vector machines(Wang, 2011b; Kazem et al., 2013), and Gaussian processes (GPs) (G.-S. Han & Lee, 2008; Yang & Lee, 2011; Park & Lee, 2012; J. Han et al., 2016). These models have also considered different types of available market information, but did not consider explicit formulation for underlying processes.

Econometric financial jump models, such as affine jump-diffusion or infinite activity Lévy processes, are alternative models that have been applied successfully for derivatives pricing and predictions (Carr et al., 2003; Kou & Wang, 2004; D. B. Madan & Seneta, 1990; D. B. Madan et al., 1998; Merton, 1976; Nualart et al., 2001; Schmitz et al., 2014). These models have been relatively successful in the valuation of contingent claims because of the ability to address volatility smile, fat tail, and volatility clustering with jumps. Econometric financial jump models, such as the CGMY or Kou models, explicitly formulate a return process of underlying assets, whereas machine learning models express the process of underlying assets implicitly from the learned model.

First of all, we propose a novel MCMC methodology based on the generative model. Similar to the particle MCMC method (Andrieu et al., 2010), the proposed method aims to acquire the sample set from posterior distribution by sampling the approximated posterior instead of estimating the exact posterior density function. In this dissertation, we suggest a generative model sampler based on variational inference and provide the theories that support the argument.

The following fundamental issues relevant in practical application will be discussed. First, In-sample estimation errors between present market and model prices calibrated from current or previous prices are compared to verify current or previous market information for each model quantitatively. Second, we measure out-of-sample prediction errors in advance for the next one day and seven days, and investigate the consistency interval of calibrated models with the market to evaluate each model based on price forecasting capability. We also consider the amount of past market information required to build each model for market prediction. Finally, the performance of domain adaptation is evaluated with the differences of in-sample training data and out-of-sample test data domains. In this empirical study, European options are used for the former training domain and American options for the latter test domain. The model should consider domain adaptation suitability for elucidating the structure of different option markets consistently with the same underlying conditions.

Conventional machine learning models are very effective in estimating the cross sectional option prices well in the data area covered by the training data, but often fail to represent the option prices outside that area. This is one of the major handicaps for applying machine learning models to option pricing and forecasting. In this chapter, we propose a generative Bayesian neural networks model for risk-neutral option pricing to overcome the limitation of conventional machine learning methods.

Lastly, we conducted a comprehensive empirical study to compare state-of-the-art American option pricing models with machine learning models with respect to model validity and model predictability for American index options using the S&P 100 index American put options from 2003 to 2012. We addressed the following fundamental questions.

- Does each model have the capability to incorporate current or previous market information well? Good fit to market prices is essential for a good model to be consistent with the markets. This fit can be verified quantitatively by comparing the in-sample estimation errors between the present market prices and the model prices calibrated from current or previous prices.

- Can each model predict future prices well? Predictability is one of the most important criteria to assess calibrated models. Predictability is evaluated by computing the out-of-sample prediction errors of the models for 1 day ahead.

- Can a machine learning model generate fair prices in the deep ITM (in the money) or deep OTM (out of the money) options as classical American option pricing models can? Capability to generate fair option prices in the domain of few transactions is a barometer for the model to elucidate consistently the financial structure of option markets.

The rest of this chapter is organized as follows. Section 2 presents the proposed generative model sampler. Section 3 evaluates the estimation and prediction performance for the American option data. Section 4 proposed the generative Bayesian neural network to overcome the limitation of machine learning methods and described the empirical experiment. Section 5 provides the conclusions of this chapter.

3.2 A Generative Model Sampler for Inference in State Space Model

3.2.1 Backgrounds

Inference in state space models

In this study, we focus on a generic state space model (SSM), a non-linear non-Gaussian hidden Markov model (HMM). A generic SSM model consists of given static parameter, $\theta \in \times$, and the following three probability distributions: an initial probability, $h(\cdot|\theta)$, a transition probability, $f(\cdot|\mathbf{x}, \theta)$, and an emission probability, $g(\cdot|\mathbf{x}, \theta)$. Figure 3.1 describes the scheme of a structure of a generic SSM.

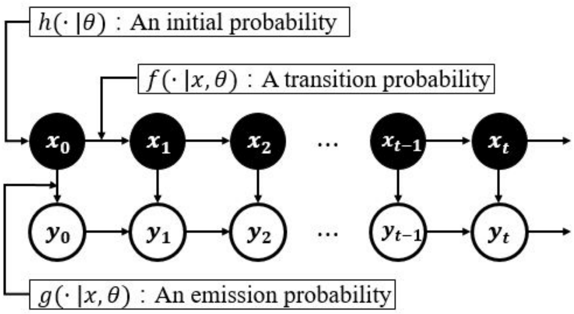


Figure 3.1 Scheme of a state space model

A generic SSM includes two types of variable, one is the visible variable $\mathbf{y} = \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_T; \mathbf{y}_i \in \mathcal{Y}^M, 0 \leq i \leq T\}$ we can observe, another is the latent variable $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T; \mathbf{x}_i \in \mathcal{X}^N, 0 \leq i \leq T\}$. Subscribe of each component means the time step. Hereafter, we shall use $\mathbf{x}$ or $\mathbf{y}$ to denote the whole process of each variable $\mathbf{x}_{0:T}$ or $\mathbf{y}_{0:T}$ for the brevity. Above mentioned three probability define the several relationships between variables under the SSM structure. A whole process of the latent variable $\mathbf{x}$ is characterized by an initial probability $\mathbf{x}_0 \sim h(\cdot|\theta)$ and a transition probability

$$\mathbf{x}_{t+1}|\mathbf{x}_t \sim f(\cdot|\mathbf{x}_t, \theta).$$

Each observation $\mathbf{y}_t$ is assumed to be conditionally independent given each latent variable $\mathbf{x}_t$ with their emission probability density:

$$\mathbf{y}_t|\mathbf{x}_t \sim g(\cdot|\mathbf{x}_t, \theta).$$

Therefore the joint density given the static parameters θ can be represented by the following form:

$$p(\mathbf{y}, \mathbf{x}|\theta) = h(\mathbf{x}_0|\theta) \prod_{t=0}^{T-1} f(\mathbf{x}_{t+1}|\mathbf{x}_t, \theta) \prod_{t=0}^{T} g(\mathbf{y}_t|\mathbf{x}_t, \theta).$$

In this context, our study aims to investigate Bayesian inference given the observed variable time series $\mathbf{y}$. We consider most general non-linear non-Gaussian SSM structure, which is hard to address the closed form for the posterior distribution: $p(\mathbf{x}|\theta, \mathbf{y})$ or $p(\mathbf{x}, \theta|\mathbf{y})$. Therefore most studies resort to approximate the posterior distribution directly or indirectly by several practical technologies such as MCMC methodologies. Further on, we suggest the novel and model-agnostic MCMC methodologies combined with a generative model to perform Bayesian inference to any general state space model. We will investigate a posterior distribution ,$p(\mathbf{x}|\theta, \mathbf{y}) \propto p(\mathbf{y}, \mathbf{x}|\theta)$ when we know parameter θ. Even the parameters θ are unknown, we also perform Bayesian inference and parameter estimation adequate to the observations $\mathbf{y}$ by evaluating a posterior distribution $p(\mathbf{x}, \theta|\mathbf{y}) \propto p(\mathbf{y}, \mathbf{x}|\theta)p(\theta)$ through our proposed methods.

Bayesian inference with MCMC methodologies

Inferences associated with state space models have been studied in a variety of approaches, such as confining to constrained models or applying to more general models with mitigated theoretical parts. In cases of linear Gaussian state space models and finite hidden Markov models, inference can be possible efficiently by sampling exactly from the posterior density with developed techniques such as a Kalman filter or constrained Gibbs sampling (Durbin & Koopman, 2002; Carter & Kohn, 1994; Fruhwirth-Schnatter, 1995). We do not discuss the case of constrained model here.

In this study, we focus on the Bayesian inference in a general state space model. Several studies have discussed the inference problem under the generic state space model, which is more suitable for applications but generally more difficult to inference (Doucet & Johansen, 2009; Poyiadjis et al., 2011; Andrieu et al., 2010). Sampling from the posterior distribution $p(\theta, \mathbf{x}|\mathbf{y})$ is a main task of Bayesian inference whereby an entire sample can be composed by alternatively updating state components $\mathbf{x}$ and stable parameters θ conditional on each other. This method is very similar to the usual Gibbs sampling method, but it has biased results because of the high dependence between latent variables $\mathbf{x}$ and parameters θ (Papaspiliopoulos et al., 2007). Recent studies have considered the sequential Monte Carlo (SMC) method to address this issue. Combined algorithm with SMC methods and MCMC approaches have been developed in the literature (Gilks & Berzuini, 2001; Andrieu et al., 2010; Chopin et al., 2013; Fulop & Li, 2013). In particular, Andrieu et al. (2010) suggested a explicit method to obtain samples from the posterior rather that the probability density value of it to overcome the main drawbacks of SMC (Fearnhead, 2002; Storvik, 2002), where SMC methods concentrate on specific particle to deteriorate the rejuvenation step as over the time steps. Estimation of the marginal likelihood for the model can be provided as a by-product by the MCMC methodologies (Chopin et al., 2013) combining the particle filtering technique and iterated batch importance sampling developed for parameter posterior evaluation by (Chopin, 2002). Fulop and Li (2013) also independently proposed a similar methodology.

Variational inference

Inference is the main algorithmic problem to account for the visible data. The range of the discussion about the data is restricted by the model assumption based on the knowledge and critical questions to data analysis. Then, we discover the pattern of the data under the restricted model through the inference. Inference answers the question: "what does the model describe about this data?" Variational inference (VI) gives general and scalable approaches to the process of inference. Consider the following probabilistic model a joint distribution of hidden variables $\mathbf{x}$ and visible variables $\mathbf{y}$

$$p(\mathbf{x}, \mathbf{y}).$$

When we want to inference about the hidden variables, the posterior distribution is the conditional distribution of the hidden variables $\mathbf{x}$ given observed variables $\mathbf{y}$

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})}.$$

A denominator of the conditional distribution is intractable for most interesting model assumptions. It causes to approximate the posterior inference.

VI brings in a variational family of distributions $q(\mathbf{x}, \nu)$ over the latent variables $\mathbf{x}$ to turn the inference into the optimization problem. For the variational parameter ν, the optimization problem finds the optimal solution $q^*(\mathbf{x}, \nu)$, which is an element of the variational family minimizing the given objective function Kullback-Leibler (KL) divergence. KL divergence is a distance between the true posterior $p(\mathbf{x}|\mathbf{y})$ with the approximated posterior $q(\mathbf{x}, \nu)$ (MacKay, 2003; Kingman, 1970):

$$\mathbf{KL}(q(\mathbf{x}), p(\mathbf{x}|\mathbf{y})) = \mathbb{E}_q \left[\log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} \right]$$

For the brevity, notation of the variational parameter ν is omitted. We cannot minimize the KL divergence exactly, but we can achieve the same goal by maximizing the evidence lower bound (ELBO), which is a lower bound on the marginal distribution $\log p(\mathbf{y})$. KL divergence can be decomposed as the following form:

$$\begin{aligned}
\mathbf{KL}(q(\mathbf{x}), p(\mathbf{x}|\mathbf{y})) &= \mathbb{E}_q \left[\log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} \right] \\
&= \mathbb{E}_q[\log q(\mathbf{x})] - \mathbb{E}_q[\log p(\mathbf{x}|\mathbf{y})] \\
&= \mathbb{E}_q[\log q(\mathbf{x})] - \mathbb{E}_q[\log p(\mathbf{x}, \mathbf{y})] + \log p(\mathbf{y}) \\
&= -(\mathbb{E}_q[\log p(\mathbf{x}, \mathbf{y})] - \mathbb{E}_q[\log q(\mathbf{x})]) + \log p(\mathbf{y})
\end{aligned}$$

Therefore, maximizing the ELBO $\mathbb{E}_q[\log p(\mathbf{x}, \mathbf{y})] - \mathbb{E}_q[\log q(\mathbf{x})]$ is equivalent to minimizing the KL divergence. General variational inference framework aims to maximizing the ELBO term efficiently based on the visible data set $\{\mathbf{y}\}_{1:T}$.

3.2.2 Proposed methods: generative model sampler

A generative model is a way of extracting samples from an undefined target density by learning the machine from available data related to the target density. A generative model could be acquired by two different approaches: a variational autoencoder (VAE) is employed when the target density has explicit relationships with other observable densities(Kingma & Welling, 2013; Kingma et al., 2014; Rezende et al., 2014), and a generative adversarial network (GAN) generates samples directly from implicit target densities(Radford et al., 2015; Goodfellow et al., 2014). Several kinds of research have been studied to obtain more accurate and meaningful samples through the combination of two approaches. Several kinds of research have been studied to obtain more accurate and meaningful samples through the combination of two approaches(Maaløe et al., 2016; Makhzani et al., 2015; Mescheder et al., 2017; Ranganath et al., 2016). In this study, we propose a modified version of the Mescheder et al. (2017)'s adversarial variational Bayes (AVB) application to extract samples from the posterior of latent variables generally difficult to clarify.

The marginal likelihood of visible variables, $\mathbf{y}$, is always greater than the sum of the expected value of the log likelihood, $p(\mathbf{y}|\mathbf{x})$, for the approximate posterior distribution, $q(\mathbf{x}|\mathbf{y})$, of latent variables, $\mathbf{x}$, and the Kullback-Leibler (KL) divergence value, which means the distance between the approximate posterior density, $q(\mathbf{x}|\mathbf{y})$, and the prior density of latent variables, $p(\mathbf{x})$, (Kingma

& Welling, 2013; Rezende et al., 2014). The following inequality shows this relationship.

$$\log p(\mathbf{y}) \geq -\mathbf{KL}(q(\mathbf{x}|\mathbf{y}), p(\mathbf{x})) + \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}[\log p(\mathbf{y}|\mathbf{x})] \tag{3.1}$$

The value of KL has the maximum value when $q(\mathbf{x}|\mathbf{y}) = p(\mathbf{x}|\mathbf{y})$, and both sides of the above inequality 3.1 become equal. Because it is practically difficult to find a $q(\mathbf{x}|\mathbf{y})$ where $q(\mathbf{x}|\mathbf{y}) = p(\mathbf{x}|\mathbf{y})$, we obtain an approximation of the posterior $p(\mathbf{x}|\mathbf{y})$ by finding the estimated distribution $q(\mathbf{x}|\mathbf{y})$ which maximizes the following equation.

$$\begin{aligned}\log p(\mathbf{y}) &= \max_q -\mathbf{KL}(q(\mathbf{x}|\mathbf{y}), p(\mathbf{x})) + \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}[\log p(\mathbf{y}|\mathbf{x})] \\ &= \max_q \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}[\log p(\mathbf{x}) - \log q(\mathbf{x}|\mathbf{y}) + \log p(\mathbf{y}|\mathbf{x})] \end{aligned} \tag{3.2}$$

Consider a real-valued discriminative networks T to circumvent the problem of calculating a probability density function value of approximated posterior $q(\mathbf{x}|\mathbf{x})$. Proposed discriminator which has the opposite way to the Mescheder et al. (2017)'s AVB enables the convergence theories of proposed MCMC methodologies proved in later section. T has the following objective function for a given $q(\mathbf{x}|\mathbf{x})$, and the sigmoid function, $\sigma(x) = (1 + \exp^{-x})^{-1}$:

$$\max_T \mathbb{E}_{p(\mathbf{x})}[\log \sigma(T)] + \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}[\log(1 - \sigma(T))] \tag{3.3}$$

Theorem 1 shows that the optimal discriminator T includes the probability density function value $q(\mathbf{x}|\mathbf{y})$ impossible to calculate practically.

Theorem 1. *The optimal discriminator T is $\log p(\mathbf{x}) - \log q(\mathbf{x}|\mathbf{y})$.*

Proof. Proof.

$$\mathbb{E}_{p(\mathbf{x})}[\log \sigma(T)] + \mathbb{E}_{q(\mathbf{x}|\mathbf{y})}[\log(1-\sigma(T))]$$
$$= \int_{\mathbf{x}} (p(\mathbf{x}) \log \sigma(T) + q(\mathbf{x}|\mathbf{y})[\log(1-\sigma(T))])$$

Under the fixed $\mathbf{x}$, the probability density function values are constant. Since the objective function is convex to σT defined between 0 and 1, the optimal is acquired from the point where the first derivative has zero value. Therefore, the optimal $\sigma(T*)$ is $\frac{p(\mathbf{x})}{p(\mathbf{x})+q(\mathbf{x}|\mathbf{y})}$. Lastly, the optimal $T*$ is $\log p(\mathbf{x}) - \log q(\mathbf{x}|\mathbf{y})$. □

Now, we can acquire the estimated marginal likelihood and samples from the approximate posterior by maximizing the objective equation 3.2 which replaces $\log p(\mathbf{x}) - \log q(\mathbf{x}|\mathbf{y})$ with $\hat{T}$ practically estimated by maximizing the objective, equation 3.3. The MCMC methodologies we propose in the next section have a similar structure to PMCMC algorithms, but have the biggest difference in replacing samples from SMC with samples from the generative model. Samples obtained by the generative model give the following characteristics to the proposed algorithm. We employs a neural networks as the generative model under the well-known fact that any continuous function can be approximated arbitrarily well by a neural network with a single hidden unit.

Using the estimate of the discriminator $T*$ obtained from the result of Theorem 1, Proposition 1 gives the estimated value of evidence $\hat{p}(\mathbf{y})$, which will be practically employed in the proposed MCMC methodologies.

Proposition 1. *The evidence* $\hat{p}\mathbf{y}$ *is estimated by* $\mathbb{E}_{\hat{q}(\mathbf{x}|\mathbf{y})}[\exp T{*}p(\mathbf{y}|\mathbf{x})]$.

Proof. Proof.

$$\begin{aligned}\log \hat{p}(\mathbf{y}) &= \mathbb{E}_{\hat{q}(\mathbf{x}|\mathbf{y})}[\log p(\mathbf{x}) - \log q(\mathbf{x}|\mathbf{y}) + \log p(\mathbf{y}|\mathbf{x})] \\ &= \mathbb{E}_{\hat{q}(\mathbf{x}|\mathbf{y})}[T* + \log p(\mathbf{y}|\mathbf{x})]\end{aligned}$$

Therefore, $\hat{p}(\mathbf{y}) = \exp \mathbb{E}_{\hat{q}(\mathbf{x}|\mathbf{y})}[T* + \log p(\mathbf{y}|\mathbf{x})]$. □

Most important achievement of embedding generative model is to implement a model-agnostic structure. We will explain in detail by the following example. Consider a situation where the latent variables, $\mathbf{x}$, of equation 3.2 composed of two subdivided groups of latent variables: $\mathbf{x}_1$ is easy to extract from the prior distribution and $\mathbf{x}_2$ is hard to extract from the prior distribution. Then the equation 3.2 can be rewritten as:

$$\log p(\mathbf{y}) = \max_q \mathbb{E}_{q(\mathbf{x}_1,\mathbf{x}_2|\mathbf{y})}[\log p(\mathbf{x}_1,\mathbf{x}_2) - \log q(\mathbf{x}_1,\mathbf{x}_2|\mathbf{y}) + \log p(\mathbf{y}|\mathbf{x}_1,\mathbf{x}_2)] \quad (3.4)$$

The whole latent variable set, $(\mathbf{x}_1, \mathbf{x}_2)$, is hard to extract from the prior distribution, $p(\mathbf{x}_1, \mathbf{x}_2)$. However, we can rewrite the prior distribution as recursive form as the above equation 3.4:

$$\log p(\mathbf{x}_1,\mathbf{x}_2) = \max_{q_2} \mathbb{E}_{q_2(\mathbf{x}_1|\mathbf{x}_2)}[\log p(\mathbf{x}_1) - \log q_2(\mathbf{x}_1|\mathbf{x}_2) + \log p(\mathbf{x}_2|\mathbf{x}_1)] \quad (3.5)$$

The above equation causes the introduction of T_2 in the same manner of that of T_1 without loss of generality. Learned T_2 is replaced to the above equation to acquire sample set, $(\mathbf{x}_1, \mathbf{x}_2)$, distributed under the prior distribution, $p(\mathbf{x}_1, \mathbf{x}_2)$. Next process is exactly the same as the basic process above: 1) Learn the discriminator T_1, and 2) Learn the posterior distribution, $q(\mathbf{x}_1, \mathbf{x}_2|\mathbf{y})$. Therefore, the proposed method can be applied to any model with many latent variables and unknown parameters because of the possibility to divide latent variables continuously in a generative model. This feature also gives real practitioner the degree of freedom to model structure for practical application.

3.3 Machine Learning versus Econometric Models in Predictability of Financial Options Markets

3.3.1 Data description and experimental design

Econometric jump and machine learning models are evaluated in terms of estimation, prediction, and domain adaptation performance by using the daily S&P 100 Index American / European put options. Two types of option domains exist: S&P 100 options with American-style exercise (ticker symbol OEX), and

S&P 100 options with European-style exercise (ticker symbol XEO). An experimental study is conducted using the S&P 100 Index American / European option data for 2012 when the effects of the recent global crisis were assumed to be maximum marginal. We considered the options with maturity from 7 to 90 days as in the literature. The option prices for very short maturity or continuing long expiration tend to be biased from low-time premium and measurement errors. The statistical summary of empirical data is demonstrated in Table 3.1. For brevity, an input variable, moneyness, is adopted as the ratio of spot price to strike price and maturity.

There are two representative econometric jump models, namely, Kou and CGMY (Kou, 2002; Carr et al., 2003) and five state-of-the-art machine learning models, including NNs, Bayesian NNs, deep NNs, SVR, and GP, for regression. The performance results of each model are evaluated based on the following widely used metrics.

(1) The mean absolute percentage error (MAPE), $\frac{1}{N}\sum_{n=1}^{N}(|e_n|/C_n^{market}$, stands for the percentage error of the model.

(2) The mean percentage error (MPE), $\frac{1}{N}\sum_{n=1}^{N}(e_n/C_n^{market})$, represents the error direction of the model.

(3) The mean absolute error (MAE), $\frac{1}{N}\sum_{n=1}^{N}|e_n|$, measures the error magnitude of the model.

(4) The root mean squared error (RMSE), $\sqrt{\frac{1}{N}\sum_{n=1}^{N}(e_n)^2}$, means the standard error of the model.

where N is the total number of options and $e_n = C_n^{market} - C_n^{model}$ is the model misspecification error where C_n^{model} is the model estimated price, and C_n^{market} is the market price for the n-th options .

Figure 3.2 shows the entire scheme of data usage for model estimation and prediction. We used 1-, 7-, and 30-day option prices for nonparametric machine learning models, and only 1-day option prices for parametric jump models for

Table 3.1 Summary statistics of the S&P 100 index American/European put options. This table reports average and standard deviation of option price with the number of observations for each category.

		Maturity							
		< 30		30 - 60		> 60		All	
moneyness		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
S&P100 index American put options by moneyness and maturities									
<0.94	price	60.52	21.34	53.13	15.99	74.38	24.56	60.64	21.48
	Observation	55		32		18		105	
0.94 - 0.97	price	28.48	5.02	31.12	4.76	35.31	4.32	29.76	5.29
	Observation	256		107		33		396	
0.97 - 1.00	price	13.18	4.28	17.72	3.62	22.63	3.47	15.86	5.05
	Observation	691		497		137		1325	
1.00 - 1.03	price	4.82	2.34	10.39	2.82	15.71	2.82	8.44	4.58
	Observation	882		773		232		1887	
1.03 - 1.06	price	2.33	1.10	5.63	2.11	10.58	2.31	4.94	3.16
	Observation	493		631		152		1276	
>1.06	price	1.62	0.59	2.56	1.39	3.98	2.44	2.78	1.80
	Observation	249		1317		488		2054	
All	price	9.72	11.61	8.58	8.52	12.07	12.15	9.53	10.41
	Observation	2626		3357		1060		7043	
S&P100 index European put options by moneyness and maturities									
<0.94	price	48.61	8.52	63.83	16.35	51.59	4.03	56.38	14.32
	Observation	10		13		4		27	
0.94 - 0.97	price	30.94	6.49	33.04	5.28	38.19	6.89	32.76	6.39
	Observation	39		39		11		89	
0.97 - 1.00	price	10.30	4.40	16.45	4.19	23.16	4.59	13.37	5.98
	Observation	305		164		44		513	
1.00 - 1.03	price	4.14	2.65	9.95	3.16	15.48	3.11	6.34	4.40
	Observation	628		254		51		933	
1.03 - 1.06	price	1.51	1.19	5.21	2.01	10.78	2.88	2.95	2.85
	Observation	464		185		32		681	
>1.06	price	0.63	0.68	2.54	1.51	4.78	2.97	1.63	1.84
	Observation	433		269		59		761	
All	price	4.47	6.57	9.73	10.17	15.23	10.87	6.81	8.80
	Observation	1879		924		201		3004	

simplicity. Unlike the machine learning models that require large amounts of data for efficient learning, parametric jump models can calibrate the model with a small amount of market data. Using the calibrated models, we compared the prediction performance of 1 day ahead and 7 days ahead, thereby generating six cases of prediction results in total. We considered the model prediction of the next 7 days in addition to the next 1 day, given that the model with a considerable predictive power for both 1-day ahead and 7-day ahead prediction is advantageous for hedging or portfolio managing purposes and for reducing inefficiency from adapting a model frequently.

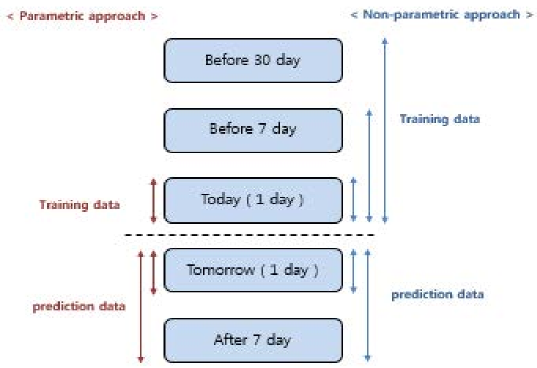


Figure 3.2 Scheme of experiments for the model estimation and prediction.

3.3.2 Estimation and prediction performance

For econometric jump models, the model is calibrated each day by estimating the parameter set that minimized the mean squared error of the actual market price and the model price calculated by the abovementioned method using OEX put option prices. A final set of calibrated parameters is obtained to be used for prediction. For machine learning models, the model is trained using the given data set (1-day, 7-day, or 30-day OEX put option prices) as stated in the previous section and obtained the final calibrated model to be used for predicting future option prices.

Table 3.2 shows the summary of estimation results for each model. Most models have acceptable estimation errors (in-sample errors), which are mostly near 10%. The estimation results of the Gaussian process model are excluded because it has practically zero estimation error by fitting exactly the option price corresponding to its moneyness and maturity with the expense of over-fitting, which resulted in poor prediction performance. Although most MPE values in

Table 3.2 Estimation performance. This table reports Estimation errors for S&P 100 OEX put options of each category.

Panel: Estimation Errors					
Model	training day	MAPE	MPE	MAE	RMSE
Kou model	1day	0.0813	0.0081	0.6013	1.1935
CGMY model	1day	0.1156	0.0081	0.7466	1.0558
NN	1day	0.1053	0.0068	0.427	1.6048
	7day	0.1232	0.0008	0.5676	1.333
	30day	0.1708	-0.0221	0.825	1.1154
BNN	1day	0.0153	0.0049	0.0513	0.3436
	7day	0.0614	-0.001	0.2743	0.3848
	30day	0.1511	-0.0255	0.7442	0.9712
SVR	1day	0.1115	-0.0337	0.9329	2.1525
	7day	0.0917	-0.0217	0.8678	2.3067
	30day	0.1332	-0.0346	1.124	3.4782

machine learning models are negative (overvalued), they are relatively small in absolute values, thereby indicating unbiased direction similar with econometric jump models. Table 5.1 presents the detailed results of model estimation errors with respect to moneyness and time to maturity. In-the-money or at-the-money options with short maturity have small estimation errors in both econometric jump and machine learning models; the latter presents no noticeable differences in maturities.

Specifically, machine learning models have small estimation errors for the region with a few observations compared with econometric jump models, which cause over-fitting in prediction. Moreover, no significant difference is observed for the estimation errors between econometric jump models using data only from previous one-day and machine learning models using data over long periods. The results of estimation partially supported the assumption that current market price generally included all information obtained previously.

Next, there are the prediction performances of each estimated model applied to out-of-sample data. The prediction results have different accuracies for each model, although most models have similar estimation errors, except for the GP model. Table 3.3 shows the prediction results of each model applied to one day and seven days ahead. Econometric jump models showed slightly better performance in one-day and seven-day predictions than machine learning models. The GP model showed the worst performance in prediction, although it showed the best estimation performance triggered by over-fitting. Machine learning models displayed mostly good prediction performance when they are trained from large option data (i.e., 30-day option prices). Interestingly, econometric jump models exhibited positive MPEs (or underpriced), whereas machine learning models showed negative MPEs (or overpriced).

Table 5.2 and 5.3 summarize the detailed prediction results of one day ahead and seven days ahead in respect to category of moneyness and maturity, respectively. For all models, the ITM or OTM options with long maturity showed large

Table 3.3 1-day & 7-day prediction performance. Panel A reports 1-day prediction errors and panel B reports 7-day prediction errors for S&P 100 OEX put options of each category.

Panel A: 1-day prediction errors					
Model	training day	MAPE	MPE	MAE	RMSE
Kou		0.1252	0.0012	0.8464	1.8336
CGMY		0.1551	0.0097	0.9406	1.3942
NN	1day	0.4801	-0.0228	2.2304	5.711
	7day	0.196	-0.0131	0.9142	2.6771
	30day	0.1851	-0.025	0.9199	1.4763
BNN	1day	0.2273	-0.0463	1.0701	3.4834
	7day	0.1446	0.003	0.6927	1.6062
	30day	0.1729	-0.0279	0.857	1.2734
SVR	1day	0.6675	-0.34	4.3862	7.9208
	7day	0.6577	-0.3239	4.4614	8.1446
	30day	0.2636	-0.0626	2.0063	5.2376
GP	1day	1.6801	-1.2734	7.358	11.0846
	7day	1.6019	-1.2107	7.2128	11.0841
	30day	0.4593	-0.2516	2.0381	5.0462
Panel B: 7-day prediction error					
Model	training day	MAPE	MPE	MAE	RMSE
Kou		0.1567	0.0146	0.9665	1.8509
CGMY		0.1872	0.0043	1.0832	1.5691
NN	1day	0.8574	-0.1391	4.0553	8.7374
	7day	0.253	-0.008	1.1586	2.9318
	30day	0.2023	-0.03	0.9965	1.5197
BNN	1day	0.3705	-0.0876	1.682	4.6454
	7day	0.2456	-0.0002	1.1185	2.3619
	30day	0.191	-0.0305	0.9542	1.3989
SVR	1day	1.0081	-0.5022	6.3333	10.6667
	7day	1.0587	-0.5473	6.5537	10.996
	30day	0.2857	-0.077	2.2053	10.2228
GP	1day	1.6517	-1.2418	7.3901	11.2432
	7day	1.5997	-1.2033	7.2809	11.1789
	30day	0.5022	-0.2699	2.1207	5.9068

relative prediction errors (MAPE and MPE). The prediction error for machine learning models increased with the volume of traded options relative to that of econometric jump models, which explained the best overall prediction performance of econometric jump models over Bayesian NNs; however, the latter showed fewer prediction errors than the former in terms of the options with long maturities of small-traded volumes. Compared with other training data, machine learning models trained by 30-day data improved the performance of the model in predicting option prices 7 days ahead, although the difference between the models using the 7-day and 30-day data is not significant in predicting option prices of the next day. By contrast, econometric jump models displayed similar range of relative prediction errors for each category of moneyness and maturity, which implied that the characteristics of return stochastic process used for the model did not change much over the time period of our interest and achieved stable performance in the 7-day and 1-day predictions.

3.3.3 Robustness and Domain Adaptation Performance of the Models

Parameters for econometric jump models and weights of machine learning models are gained through the estimation step. Weights of machine models can be regarded as parameters which provide intact models without empty parameters from the given market data like as parameters of econometric jump models do. the hypothesis is assumed that well defined parameter from the estimated model has only slight changes after every day update as long as the absence of significant changes in the market. In this sense, the robustness of parameters means that a set of the daily calibrated parameters or weights for a model is confined to a relatively small region.

Given that the calibrated weights for a machine learning model are highly dimensional, A multidimensional scaling method (MDS) Lattin et al. (2003) is used to visualize the proximities of parameters or weights of each model. Multidimensional scaling is a widely used dimension reduction method that transforms a set of high-dimensional observations into a set of low-dimensional observations by approximately preserving the distances or dissimilarities between all pairs of observations.

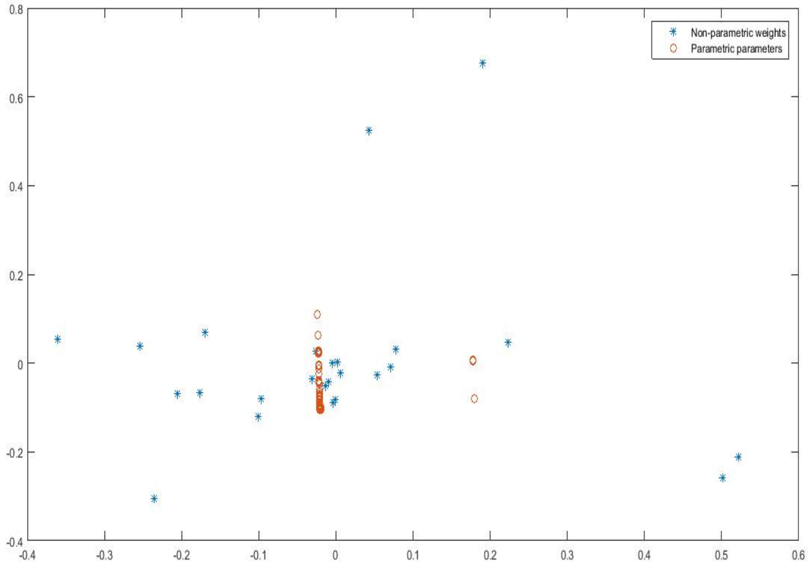


Figure 3.3 Two dimensional MDS visualization of each parameters. Red 'o': corresponding 1-day parametric parameters. Blue '*': corresponding 1-day non-parametric parameters.

To illustrate our results, MDS is applied to the calibrated parameters of the Kou model and to the calibrated weights of the Bayesian NN model; the dimensions of the two are 4 and 246, respectively. The constructed 2D MDS visualizes the 2D locations of daily parameters or weights of each model. Figure 3.3 shows a typical plot for the 2D MDS visualization of the econometric jump model and the machine learning model. The two types of models present a different trend. The Kou model parameters, represented by red "o"s, are mostly confined to a small range of regions. The Bayesian neural network model weights, represented by blue "*"s, are widely scattered with no noticeable patterns. The Bayesian neural network model weights, represented by blue "*"s, are widely scattered with no noticeable patterns. This result implies that econometric jump model parameters are more stable and robust than machine learning model parameters.

As one of the transfer learning in machine learning fields, domain adaptation aims to learn in the test domain, which is not used in training, with the information in training domain(Ben-David et al., 2010; Pan & Yang, 2010). Studies in this area have been conducted in such a way that source knowledge distributions are adjusted in a manner similar to new target knowledge distributions (Gong et al., 2012; Patricia & Caputo, 2014).

We compared the domain adaptation performance of the models by predicting American put option prices of the following day using models calibrated from current European put option prices. Different types of options reflected different demands and interests of investors obtaining from different payoffs and acquired information. The requirement for domain adaptation in option markets occurs naturally when different types of options have the same underlying assets. Hence, a model calibrated from one type of options should be adapted to predict another type of options.

Table 3.4 1-day domain-adaptation performance. This table reports 1-day domain-adaptation errors of each model. Each model is trained by European S&P 100 XEO put options and tested by American S&P 100 OEX put options.

Model	MAPE	MPE	MAE	RMSE
Kou model	0.1517	-0.0244	1.0418	1.8085
CGMY model	0.1722	-0.3082	2.2802	2.3055
NN	0.5425	-0.0046	2.3463	5.4485
BNN	0.1849	-0.0927	0.9867	3.3072
SVR	0.6683	-0.3398	4.3999	7.9467
GP	1.5873	-1.2362	6.8157	10.574

Table 3.4 shows that econometric jump models exhibit better domain adaptation performance than machine learning models, although all the models show worse performance with different domains than with the same domain as expected. Notably, the performances of econometric jump models with different domains are still better than those of machine learning models with the same domain. Thus, we may use domain adaptation algorithms, such as sample selection bias in covariate shift, learning shared representations, or feature-based supervised adaptation, (see (Ben-David et al., 2010; Pan & Yang, 2010) and the references therein for more details) for machine learning models to enhance performance. However, theoretically, prediction performance using different domains cannot be better than that using the same domain, (see the proof in (Ben-David et al., 2010; Pan & Yang, 2010)); thus, econometric jump models are superior to machine learning ones in terms of domain adaption performance.

In addition, the relatively small prediction errors of the parametric models adopting different domains show that their underlying risk-neutral dynamics of returns provide suitable and consistent models to explain the two different types of option markets well. The results of domain adaptation takes into account different fundamental approaches of two categories; the existence of explicit form of the underlying process. In case of econometric jump models, the explicit underlying process, such as kou model, plays the role as a bridge between two domains. Model parameters from one domain transform the information compatible to the other domain by adapting explicit underlying process. On the other hand, machine learning methods without intermediate factors have to employ further techniques which adjust distributions between domains (Gong et al., 2012; Patricia & Caputo, 2014). For financial derivative pricing purposes, domain adaptation is possible without the introduction of additional technologies under the explicit underlying process.

Table 5.4 shows the detailed domain adaptation results of 1-day-ahead prediction in each category of moneyness and maturity. The result is similar to those of 1-day-ahead and 7-day-ahead prediction errors, but more dramatic.

3.4 A Generative Bayesian Neural Networks Model for Risk-Neutral Option Pricing

3.4.1 Proposed method

Figure 3.4 presents the daily option prices traded with the 15 expiration date. In general, the options in the extreme area defined above are not traded frequently. Figure 3.4 shows that the pricing methodology based on the CGMY model provides consistent price estimates, but Bayesian NN has poor prediction performance in the extreme domain (i.e., the region where there are no currently actual transactions). The reason why the Bayesian NN fails to give a consistent shape is that there is no data to learn at the extreme region. It is the main drawback of conventional machine learning models that they almost always fail to represent the area with few data.

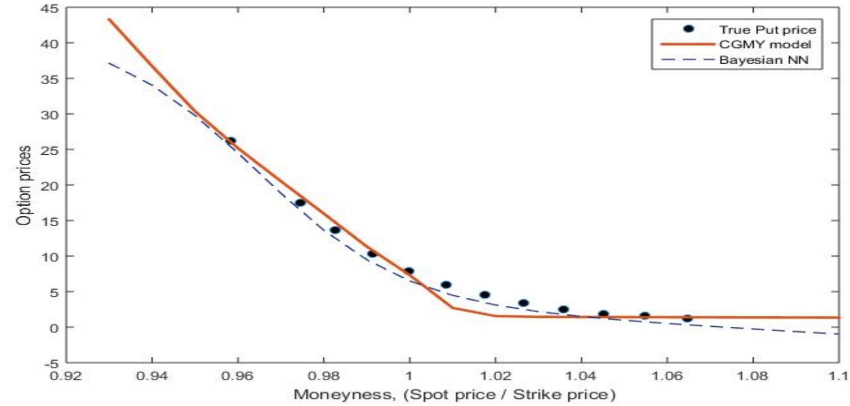
To overcome this problem, we propose a generative Bayesian learning model with a prior incorporating a financial structure such as law of one price as follows. Given a data set $\mathcal{D} = \{(\mathbf{x}_i^t, y_i^t)|\ i = 1, ..., n_t,\ t = 0, 1, ..., \ell\}$, we assume that the conditional distribution of the output option value y^t at time t is given by

$$y_i^t = f(\mathbf{x}_i^t, \mathbf{w}) + \epsilon_i,\ \forall i = 1, ..., n_t, \tag{3.6}$$

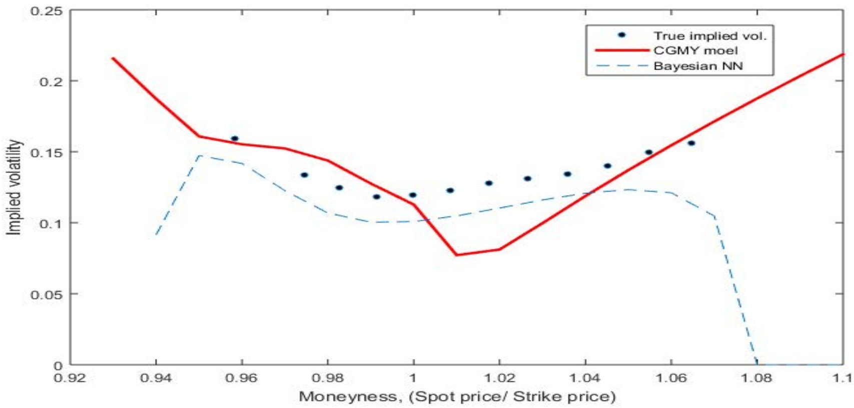
$$\text{or equivalently,} \qquad p(\mathcal{D}|\mathbf{w}) = \prod_{i=1}^{n_t} \mathcal{N}(y_i^t - f(\mathbf{x}_i^t, \mathbf{w}), \sigma^2) \tag{3.7}$$

where $f(\mathbf{x}^t, \mathbf{w})$ is a neural network model with weight vector $\mathbf{w}$ to be estimated and ϵ_i is an additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ with mean zero and variance σ^2 arising from market frictions. We define a generative prior probability distribution over the weight vector $\mathbf{w}$ at time t as

$$p(\mathbf{w}|\mathbf{w}^{t-1}) = \prod_{k=1}^{\nu} \mathcal{N}(f(\mathbf{x}_k, \mathbf{w}) - f(\mathbf{x}_k, \mathbf{w}^{t-1}), \sigma_0^2), \tag{3.8}$$



(a)



(b)

Figure 3.4 (a) Put option prices and (b) implied volatilities according to the moneyness with the same expiration date estimated from the CGMY model and the Bayesian Neural Network model.

which implies that under this prior, the new option values at some deep ITM or deep OTM samples $\mathbf{x}_k$, $k = 1, ..., \nu$ are similar to the previous option values $f(\mathbf{x}_k, \mathbf{w}^{t-1})$ up to an additive Gaussian noise with mean zero and variance σ_0^2.

The Bayes's rule leads to the posterior probability distribution over the weight vector $\mathbf{w}$ at time t given by

$$\begin{aligned}
&\ln p(\mathbf{w}|\mathcal{D}) = \ln p(\mathcal{D}|\mathbf{w}) + \ln p(\mathbf{w}|\mathbf{w}^{t-1}) - \ln p(\mathcal{D}) \\
&= \sum_{i=1}^{n_t} \ln \mathcal{N}(y_i^t - f(\mathbf{x}_i^t, \mathbf{w}), \sigma^2) + \sum_{k=1}^{\nu} \ln \mathcal{N}(f(\mathbf{x}_k, \mathbf{w}) - f(\mathbf{x}_k, \mathbf{w}^{t-1}), \sigma_0^2) + const. \\
&= -\frac{1}{2\sigma^2} \sum_{i=1}^{n_t} (f(\mathbf{x}_i^t, \mathbf{w}) - y_i)^2 - \frac{1}{2\sigma_0^2} \sum_{k=1}^{\nu} (f(\mathbf{x}_k, \mathbf{w}) - f(\mathbf{x}_k, \mathbf{w}^{t-1}))^2 + const.
\end{aligned}$$

The maximum a posteriori (MAP) estimator for the weight vector $\mathbf{w}$ can then be equivalently achieved by finding the minimum value of the following objective function

$$E(\mathbf{w}) = \sum_{i=1}^{n_t} (f(\mathbf{x}_i^t, \mathbf{w}) - y_i)^2 + \lambda \sum_{k=1}^{\nu} (f(\mathbf{x}_k, \mathbf{w}) - f(\mathbf{x}_k, \mathbf{w}^{t-1}))^2 \tag{3.9}$$

where $\lambda = \sigma^2/\sigma_0^2$ is a user-controlled parameter and ν is the number of prior samples. The second term in equation (3.9) represents a penalty term imposing the model does not fluctuate from the previous updates. Appropriate prior samples are generated from the previous model in a way that avoids the pricing bias for the extreme ITM or OTM options rarely traded and maintains the distribution of trading frequency. They are then used to train the model by augmenting the training data.

Regarding to the initial choice of the weight vector $\mathbf{w}^0$ at time $t = 0$, we prefer to train the model using an artificial sample generated from a risk-neutral financial option model such as CGMY model to guarantee the no-arbitrage conditions for deep ITM or OTM options. Then we update the weight vector $\mathbf{w}^t$ at the next time $t > 0$ by using both the training data at time t and some prior samples for the deep ITM and deep OTM option data that are simulated from the GBNN at time $t - 1$. We'd like to achieve the following goals in the proposed learning algorithm. First, the proposed model is expected to learn reasonable prices that satisfy economic conditions such as no-arbitrage by incorporating prior information obtained from risk-neutral financial option models. Second, the proposed method is expected to self-evolve by tuning to an updated prior samples generated on the previous machine. The entire procedure of the proposed method is summarized in Algorithm 1.

Algorithm 1 Generative Bayesian neural network (GBNN)

Require: Given a data set $\mathcal{D} = \{(\mathbf{x}_i^t, y_i^t)|\ i = 1, ..., n_t,\ t = 0, 1, ..., \ell\}$; set ν the number of prior samples. For the initial prior sampling, generate ν-prior samples $\mathcal{S}^0 = \{(\mathbf{x}_k, f(\mathbf{x}_k; \mathbf{w}^0))|\ k = 1, ..., \nu\}$ where $f(\mathbf{x}_k; \mathbf{w}^0))$ is the option values predicted by the risk-neutral financial model such as CGMY model.

Ensure: trained GBNN $f(\mathbf{x}; \mathbf{w})$

```
for t = 1 : ℓ do
    {Step 1} Prior sampling
    Generate ν-prior samples S = {(x_k, f(x_k; w^{t−1}))| k = 1, ..., ν} from the
      GBNN f(x; w^{t−1}) where x_k is the pair of moneyness and maturity for
      the deep ITM and the deep OTM options a user provided.
    {Step 2} Learning the GBNN
    for t = 1 : n_t do
        Input D_t = {(x_i^t, y_i^t)| i = 1, ..., n_t} ∪ {(x_k, f(x_k; w^{t−1}))| k = 1, ..., ν}
        Output Train GBNN f(x; w) using an augmented data set D_t and set
          the new updated weight vector as w^t.
    end for
end for
```

Figure 3.5 shows that the proposed GBNN fits well to the extreme ITM or OTM option prices and self-evolves consistently with that of financial option models at those extreme options for the next three months with no corrections.

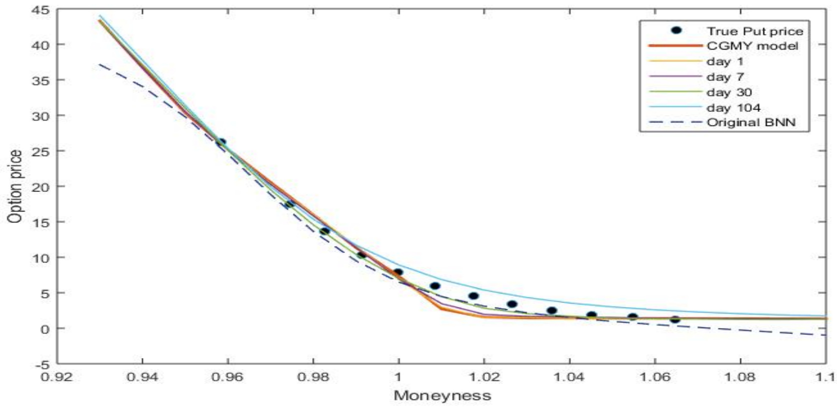


Figure 3.5 Evolving process of proposed machine learning models through the times

3.4.2 Empirical Studies

We compared the models introduced in the previous sections in terms of their capabilities for calibration, and prediction using the S&P 100 index American/European put options. First, we described the data for the empirical studies. Second, we compared the in-sample estimation errors of each models to evaluate their validity and the out-of-sample prediction errors using the calibrated models to verify their predictive power for option pricing.

Summary of the data

We used daily market data from the S&P 100 index options. The S&P 100 index is a weighted stock market index of the largest and most established 100 companies in the S&P 500 updated by Standard & Poor's. The S&P 100 index option contract has an underlying value that is equal to the value of the S&P 100 index and offers two different types of option domains: S&P 100 options with American-style exercise (ticker symbol OEX) and S&P 100 options with European-style exercise (ticker symbol XEO). Since 1983, investors

have used OEX to adjust their equity portfolio exposure, and more than one billion OEX options have been traded. In July 2001, CBOE introduced cash-settled S&P 100 options (ticker symbol XEO) with European-style exercise. The exercise-settlement value is calculated using the reported closing sales price in the primary market of each component stock on the last business day before the expiration date or on the day the expiration notice is properly submitted if exercised before expiration. We used the OEX put option data traded from 2003 to 2012 for our experiments. According to CBOE reports, OEX options are considerably more actively traded during whole periods than XEO options, and put option contract volume is considerably larger than that of the call option.
We used simple moneyness, κ, which is the ratio of spot price to strike price, to describe the relative position of the present price of an underlying to the strike price of an option. the moneyness used for the empirical analysis ranges from 0.4909 to 1.8568 as the maturity and trading day changes. We performed the conventional data pre-processing step in literature to eliminate distortion in the experiment. Options with less than 7 or more than 90 days to expiration were removed from the data. Short time-to-maturity τ tends to cause distortion because of low time premium and bid-ask spread; meanwhile, long expiration may cause biases and measurement errors. The summary statistics, such as average price and standard deviation of OEX options in accordance with time-to-maturity and moneyness, have been provided in Table 3.5.

The average option price over the period is 9.53 for American-style OEX options. The number of observations is large for the ATM options (moneyness range of 0.97-1.03) with short maturity and small for OTM options (moneyness greater than 1.03) with 30-60 days maturity or longer. The variances of option prices traded in the money (ITM) are relatively large while they decreases as simple moneyness κ increases. Table 3.6 shows the trading volumes of OEX options. The total volume has been significantly increased during the financial crisis especially for OTM and ITM options.

Table 3.5 Summary statistics of the S&P 100 index American put options from 2003 to 2012. This table reports average and standard deviation of option price with the number of observations for each categories.

		Maturity							
		< 30		30 - 60		> 60		All	
moneyness		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
S&P100 index American put options by moneyness and maturities									
<0.94	price	63.87	36.98	69.53	41.29	80	45.99	68.68	40.62
	Obs.	2076		1824		744		4644	
0.94 - 0.97	price	27.12	5.63	30.08	6.02	34.14	7.08	29.18	6.42
	Obs.	2858		2390		751		5999	
0.97 - 1.00	price	13.2	4.88	17.42	5.67	21.31	6.73	16.22	6.24
	Obs.	6129		6189		2179		14497	
1.00 - 1.03	price	5.57	3.67	10.34	5.12	14.29	6.22	9.33	5.85
	Obs.	6255		7618		3205		17078	
1.03 - 1.06	price	3.56	2.91	6.27	4.35	9.56	5.38	6.13	4.71
	Obs.	3880		6718		2644		13242	
>1.06	price	2.76	2.34	3.65	3.07	4.9	3.93	3.85	3.31
	Obs.	4829		16283		8094		29206	
All	price	13.56	19.94	11.87	17.28	13.75	19.09	12.78	18.53
	Obs.	26027		41022		17617		84666	

We considered three classical financial models, the CGMY model, the Heston model, and the GARCH(1,1) model. To acquire the American option prices, we selected several pricing methods, namely, the least squares Monte Carlo method (LSM), the Barone-Adesi Whaley methods (BW), the ad-hoc Black-Scholes model (AH-BS), the ad-hoc local volatility model (AH-LV) to compare the performance of them with those of state-of-the-art machine learning models such as the Bayesian neural networks (BNN), the support vector regression (SVR), the Gaussian processes (GP), and the generative Bayesian neural networks (GBNN). We evaluated the performance result of each model according to the four widely used metrics.

(1) The mean absolute percentage error (MAPE), $(\sum_{n=1}^{N} |\varepsilon_n|/C_n^{mkt})/N$, stands

Table 3.6 Trading volumes of S&P index American put options for each year in respect to the moneyness(κ) and time to maturities(τ). We divide moneyness into three ranges: ; (1) ITM(In the money), $\kappa < 0.97$; (2) ATM(At the money), $0.97 \leq \kappa < 1.03$; (3) OTM(Out of the money), $\kappa \geq 1.03$.

Panel A: Trading volume by κ				Panel B: Trading volume by τ			
year	ITM	ATM	OTM	τ <30	30-60	τ >60	All
2003	1111	2861	4551	2391	4032	2100	8523
2004	608	3171	3102	1786	3332	1763	6881
2005	530	3247	2183	1510	2845	1605	5960
2006	377	3538	2337	1496	3011	1745	6252
2007	815	3897	4549	2699	4531	2031	9261
2008	2680	3164	6362	4313	5694	2199	12206
2009	1661	2488	5848	3031	4990	1976	9997
2010	1381	2965	5052	2891	4711	1796	9398
2011	979	3045	5145	3299	4525	1345	9169
2012	501	3199	3319	2611	3351	1057	7019

for the percentage error of the model.

(2) The mean percentage error (MPE), $(\sum_{n=1}^{N} \varepsilon_n / C_n^{mkt})/N$, represents the error direction of the model.

(3) The mean absolute error (MAE), $(\sum_{n=1}^{N} |\varepsilon_n|)/N$, measures the error magnitude of the model.

(4) The root mean squared error (RMSE), $\sqrt{(\sum_{n=1}^{N} (\varepsilon_n)^2)/N}$, means the standard error of the model.

where N is the total number of options and $\varepsilon_n = C_n^{mkt} - C_n^{model}$ is the model misspecification error where C_n^{model} is the model estimated price, and C_n^{mkt} is the market price.

Estimation performance

We investigated whether classical financial models and machine learning models can be estimated to a given market data well, which is a prerequisite for a good model to be consistent with the current market information. We used the one day option prices to calibrate each model and calculated its in-sample error for each day. Then we reported the estimation errors with three different time domains: pre-crisis from 2003 to 2006, financial crisis from 2007 to 2009, post-crisis from 2010 to 2012.

Table 3.7 shows a summary of the estimation results of each model. Most models have acceptable estimation errors (in-sample errors) mostly near 10% based on MAPE. We excluded the estimation result of the Gaussian process model in this table because it has almost zero estimation error at the expense of over-fitting, often resulting in poor prediction performance. For the ITM options during the financial-crisis in panel B, some machine learning models such as support vector regression and Bayesian neural networks show relatively large estimation errors, partly due to extrapolated option prices. In contrast, a generative Bayesian neural network overcomes such problem by adapting to no-arbitrage conditions. The machine learning models show relatively larger estimation errors during the financial crisis period than those of the classical American option pricing models. In the estimation phase, Weird or unusual market situation makes the former react more actively than the latter.

Table 5.5 presents the detailed results of model estimation errors with respect to moneyness and time to maturity for all of the four evaluation measures. In-the-money or at-the-money options with short maturity have small estimation errors in both the classical models and machine learning models; the latter presents no noticeable differences in maturities. Notably, the machine learning models have small calibration errors for the region with a few observations unlike the parametric jump models, which cause over-fitting in prediction.

Prediction performance

The differences in the overall estimation errors between the machine learning models and classical American option pricing models are not significant, except during the financial crisis period. We then examined the prediction performance

Table 3.7 Estimation performance. This table reports MAPE and RMSE for S&P 100 index American put options of each categories with respect to the ratio of the spot to the strike prices. We divide moneyness into three ranges: ; (1) ITM(In the money), $\kappa < 0.97$; (2) ATM(At the money), $0.97 \leq \kappa < 1.03$; (3) OTM(Out of the money), $\kappa \geq 1.03$. GBNN is the generative Bayesian neural networks; BNN is the Bayesian neural networks; SVR is the support vector regression; CGMY is the estimated result under the CGMY model; AH-BS is the ad-hoc Black-Scholes model; AH-LV is the ad-hoc local volatility model.

Panel A: Estimation error during the pre-crisis, from 2003 to 2006.

	MAPE				RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0409	0.1098	0.199	0.1426	2.1862	1.4548	1.0964	1.4378
BNN	0.0506	0.1863	0.3418	0.2419	2.268	1.3173	1.2345	1.4264
SVR	0.2171	0.1965	0.2024	0.201	9.1274	1.9676	0.6201	3.487
CGMY	0.0649	0.1344	0.2524	0.1789	2.3526	1.786	1.0893	1.6081
AH-BS	0.054	0.0849	0.2288	0.1457	1.9092	1.0538	0.8097	1.0753
AH-LV	0.0538	0.0853	0.2308	0.1468	1.9066	1.0584	0.817	1.0791

Panel B: Estimation error during the financial-crisis, from 2007 to 2009.

	MAPE				RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0403	0.1269	0.177	0.1385	4.9511	3.5672	2.5193	3.4489
BNN	0.0464	0.1629	0.4304	0.2861	5.219	2.0649	2.0744	3.0254
SVR	0.1537	0.1499	0.202	0.1783	24.372	2.5602	0.9702	11.8373
CGMY	0.0435	0.0999	0.2013	0.1388	2.6952	1.8563	1.4546	1.8919
AH-BS	0.0383	0.0639	0.2203	0.1435	2.5082	1.438	1.3075	1.6146
AH-LV	0.0381	0.0631	0.2199	0.143	2.4865	1.4425	1.2997	1.6057

Panel C: Estimation error during the post-crisis, from 2010 to 2012.

	MAPE				RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0345	0.1045	0.1813	0.1338	2.6295	1.7108	1.4304	1.7562
BNN	0.0531	0.1833	0.3916	0.2788	2.7483	1.6744	1.5986	1.7962
SVR	0.212	0.198	0.2127	0.2068	13.057	2.5955	0.8212	4.8883
CGMY	0.07	0.1423	0.2677	0.2011	3.1047	2.3552	1.4561	2.0577
AH-BS	0.0391	0.0686	0.2469	0.1596	1.9105	1.2545	1.0699	1.267
AH-LV	0.0393	0.0692	0.2482	0.1605	1.9007	1.2556	1.0701	1.2644

of each trained model applied to out-of-sample data by comparing its predictive performance of 1 day ahead. GBNN is the generative Bayesian neural networks;

BNN is the Bayesian neural networks; SVR is the support vector regression; GP is the Gaussian processes; CGMY is the prediction results under the CGMY model; LSM-BS is the least squares Monte Carlo (LSM) approach under the Black-Scholes model; LSM-GARCH is the least squares Monte Carlo (LSM) approach under the GARCH model; LSM-Heston is the least squares Monte Carlo (LSM) approach under the Heston model; BW is the Barone-Adesi Whaley methods; AH-BS is the ad-hoc Black-Scholes model; AH-LV is the ad-hoc local volatility model. Table 3.8 shows the prediction results of each model.

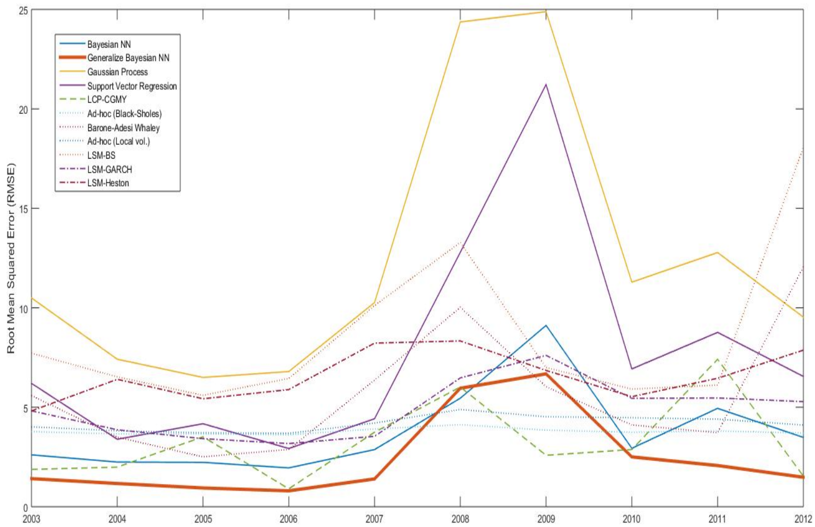


Figure 3.6 The yearly total RMSE performance of each models.

Generative Bayesian neural networks outperformed the other models in overall prediction accuracy as shown in Figure 3.6. The GBNN also shows a quite robust performance compared with other machine learning models. Notably, all the models show relatively large prediction errors during the financial crisis period. The GP model shows the worst performance in prediction, although it shows the best calibration performance triggered by over-fitting.

Table 3.8 Prediction performance. Table reports MAPE and RMSE for S&P 100 index American put options of each categories with respect to the ratio of the spot to the strike prices. We divide moneyness into three ranges: ; (1) ITM(In the money), $\kappa < 0.97$; (2) ATM(At the money), $0.97 \leq \kappa < 1.03$; (3) OTM(Out of the money), $\kappa \geq 1.03$.

Panel A: Prediction error during the pre-crisis, from 2003 to 2006.

	Panel A: MAPE				Panel B: RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0477	0.1139	0.206	0.149	2.2307	1.1336	0.728	1.1663
BNN	0.1016	0.2066	0.3557	0.2623	5.2196	1.6207	1.2922	2.3009
SVR	0.251	0.2802	0.5241	0.3849	11.6961	2.6816	1.5816	4.746
GP	0.4861	0.5031	2.0351	1.174	19.3942	4.6159	5.5548	8.1979
CGMY	0.064	0.3133	0.4936	0.3724	2.2297	2.7024	1.4492	2.2149
LSM-BS	0.3606	0.56	0.7895	0.6422	12.8136	6.7059	4.1986	6.721
LSM-GARCH	0.0737	0.3235	0.5937	0.4189	3.0750	3.4207	2.9134	3.1763
LSM-Heston	0.2966	1.3449	1.7322	1.4159	8.9468	8.3623	4.7188	7.0627
BW	0.1007	0.3619	0.6855	0.4795	4.033	4.5893	3.1919	3.9865
AH-BS	0.1059	0.2869	0.6528	0.4132	6.4595	2.6646	2.2092	3.5679
AH-LV	0.1115	0.2845	0.642	0.4063	6.8498	2.7997	2.248	3.6749

Panel B: Prediction error during the financial-crisis, from 2007 to 2009.

	Panel A: MAPE				Panel B: RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0608	0.1375	0.2373	0.1767	10.0039	3.9517	2.94	5.4924
BNN	0.0903	0.189	0.4562	0.3148	12.6347	2.5384	2.2733	6.3454
SVR	0.1985	0.261	0.6152	0.4394	30.1629	4.1149	2.6394	14.7507
GP	0.4387	0.3911	2.8442	1.6998	41.2363	6.6496	11.8471	21.3966
CGMY	0.1677	0.6515	0.5604	0.5568	5.823	7.5397	2.9603	5.8118
LSM-BS	0.3163	0.5622	0.8186	0.6585	17.9526	11.4083	6.2008	10.6835
LSM-GARCH	0.1028	0.3906	0.8612	0.5941	6.5499	8.3248	6.3366	7.0857
LSM-Heston	0.1764	0.8209	1.0179	0.8202	8.9899	9.2932	5.5659	7.4803
BW	0.1328	0.4731	0.8105	0.597	7.4988	10.0874	6.3224	7.9046
AH-BS	0.088	0.2991	0.8133	0.5383	4.9226	4.1446	3.5178	3.9758
AH-LV	0.0913	0.3062	0.8115	0.54	6.4445	4.7103	3.6754	4.5792

Panel C: Prediction error during the post-crisis, from 2010 to 2012.

	Panel A: MAPE				Panel B: RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.045	0.1142	0.2391	0.1667	3.6036	1.8372	1.6229	2.1386
BNN	0.1006	0.2099	0.4266	0.315	8.4125	2.3171	1.9041	4.0556
SVR	0.2611	0.318	0.693	0.5097	18.5397	4.0382	2.5583	7.917
GP	0.4636	0.4595	2.255	1.4046	25.5683	5.8726	7.0698	11.4126
CGMY	0.0628	0.3484	0.5471	0.425	2.461	3.5599	2.1684	2.8416
LSM-BS	0.191	1.4579	2.4174	1.8232	8.8681	13.3928	8.7525	10.7795
LSM-GARCH	0.0632	0.2463	0.7562	0.4868	3.3124	4.3034	4.1397	4.1512
LSM-Heston	0.2307	1.2878	1.3858	1.2354	9.0887	9.3133	4.8895	7.3842
B-W	0.1083	0.9957	1.4524	1.1378	4.7103	9.1892	5.6676	7.161
AH-BS	0.1048	0.3475	0.8959	0.6101	4.517	4.0779	3.3299	3.7606
AH-LV	0.1109	0.3434	0.8931	0.6078	6.868	4.3137	3.4275	4.3508

Table 3.8 shows that the GBNN methods has better prediction performance for OTM and ATM options than the CGMY methods except for ITM options. For instance, the GBNN method shows the corresponding ITM, ATM, OTM prediction accuracies are 3.6036, 1.8372, and 1.6229 for the post-crisis period. The CGMY model shows that the corresponding accuracies are 2.461, 3.5599, and 2.1684 for the same period. When considering the trading volume of ITM options is not large in the market as a whole as can be seen in Table 3.6, the high accuracy of the ATM and OTM regions is a prominent advantage of the GBNN. Based on the MAPE measure, the predictability of GBNN model is superior to that of the CGMY model. Because MAPE measures the ratio of error considering the size of the price, MAPE is a suitable for evaluating American option prices that range widely like from \$1 to \$100.

Given the stochastic volatility process, the LSM methods under the GARCH(1,1) and Heston model shows improved prediction accuracies than the LSM under the BS model. Albeit in considering stochastic volatilities, prediction accuracies under the stochastic volatility models are lower than the CGMY model. This seems to be due to the calibration process of the stochastic volatility model, which utilizes only the return asset and European options, unlike that of the CGMY model which considers the American option. In particular, the Heston model has worse performance than other stochastic volatility model, the GARCH(1,1) model. It may be caused from that the complicated calibration process has several local solution problem by introducing correlation parameter between return and volatility processes.

Table 5.6 summarizes the detailed prediction results in each category of moneyness and maturity, respectively. The machine learning models show a trend of decreasing prediction errors as the moneyness increases, while the classical American option models show no discernible trend. Lower prediction accuracy of the machine learning models for the ITM options is partly due to the fewer observations to train the models.

We measured the forecasting performance of future 7-day based on the data which is used for estimating the model. We have excluded other methodologies

Table 3.9 Prediction performance for 7-day ahead. Table reports MAPE and RMSE for S&P 100 index American put options of each categories with respect to the ratio of the spot to the strike prices. We divide moneyness into three ranges: ; (1) ITM(In the money), $\kappa < 0.97$; (2) ATM(At the money), $0.97 \leq \kappa < 1.03$; (3) OTM(Out of the money), $\kappa \geq 1.03$. GBNN is the generative Bayesian neural networks; CGMY is the prediction results under the CGMY model.

Panel A: Prediction error during the pre-crisis, from 2003 to 2006.

	Panel A: MAPE				Panel B: RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0505	0.1362	0.2784	0.1902	4.5525	1.4462	1.1876	1.8843
CGMY	0.1649	0.5631	1.0272	0.7298	6.6954	6.7178	4.7920	5.9439

Panel B: Prediction error during the financial-crisis, from 2007 to 2009.

	Panel A: MAPE				Panel B: RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0859	0.2445	0.4328	0.3182	10.6135	4.1154	3.5251	5.5029
CGMY	0.1576	0.6724	0.6266	0.5636	6.8005	8.5670	5.3630	6.7210

Panel C: Prediction error during the post-crisis, from 2010 to 2012.

	Panel A: MAPE				Panel B: RMSE			
Model	ITM	ATM	OTM	All	ITM	ATM	OTM	All
GBNN	0.0555	0.1460	0.3417	0.2386	6.5430	2.7443	2.3762	3.2803
CGMY	0.1158	0.4404	0.6487	0.5145	5.0639	5.8230	4.5005	5.1183

that showed relatively lower accuracy in the 1-day ahead prediction than the GBNN and CGMY models. Table 3.9 shows that the GBNN methods actually slightly outperform the CGMY model for the each range. Table 5.7 summarizes the detailed prediction results in each category of moneyness and maturity, respectively. The prediction results suggest that the parameters of the GBNN method are appropriately estimated given that the 7-day ahead out-of-sample performance may actually turn out to be fairly satisfactory. As an out-of-sample window is longer, the velocity of decreasing prediction performance is faster for the CGMY model than the GBNN model.

We also reveal that the proposed GBNN method has advantages over the CGMY model in terms of computation time and model consistency to highlight the suitability of the proposed GBNN for practical application. We have demonstrated that the calibration and pricing time for the GBNN model, the Fourier cosine method and LCP method under the CGMY model. The computer used for all experiments has an Intel(R) Core(TM) i7-4820K CPU @ 2.70GHz with 64.0 GB; The code is written in MATLAB R2017a.

We employed the 4-point Richardson extrapolation on Bermudan puts with 512 series expansion and 5 monitoring dates to approximate American put options by using Fang and Oosterlee methods. We also considered 1024 time and 40 spatial steps for the LCP pricing methods. In the Calibration phase, the LCP and Fourier Cosine methods take more than two hours to estimate the parameter sets while the GBNN model is trained within one minute given the daily data. This result seems to be caused from that the nonlinear optimization procedure included in the calibration process using the American options. The CGMY model assumed that the volatility process is deterministic, which also has a small number of parameters available to account for the process of underlying. However, it seems that the constrained model with the meaningful parameters makes it difficult to optimize the suitable model for the market data. One way to overcome these drawbacks is to solve the easier optimization problem by mitigating the restrictions of the model. For example, there is a stochastic volatility model considering volatility as a stochastic variable and a local volatility model, which regards the market volatility as a conditional expectation of an instantaneous volatility (Gatheral, 2011). The GBNN model alleviates the constraints of the model by as much as the machine learning while trying to derive from the data the assumptions such as the no-arbitrage assumption that can be obtained from the model. It seems to be suitable for the practical purposes, such as to learn for predicting the next day option prices, since the GBNN has shorter calibration time than other methods under the CGMY model. Corresponding pricing times of the GBNN, COS-CGMY, LCP-CGMY are 0.01349, 0.3012, and 0.2975 seconds. There is no significant difference in pricing time between models. A similar pricing time for the LCP and COS methods seems to have resulted from requiring four independent Bermudan option pricing for American options with the Fourier cosine method.

Table 3.10 The mean and standard deviation of RMSE for 1-day and 7-day prediction given 50 independent trial. Table reports mean and standard deviation of RMSE for 1-day and 7-day prediction of American put options from 50 independent trial. GBNN is the generative Bayesian neural networks; LCP-CGMY is the LCP methods under the CGMY model.

	1-day prediction		7-day prediction	
Model	Mean	stdev.	Mean	stdev.
GBNN	1.0094	0.1870	1.4312	0.1324
LCP-CGMY	2.3320	2.3404	3.5937	2.7689
COS-CGMY	2.5299	3.9868	3.1828	2.9136

We also have compared the consistency of the estimated parameters for the prediction from the additive empirical analysis. We performed 50 independent calibrations using the same data and acquired 1-day and 7-day ahead prediction results from the 50 estimated parameter sets, respectively. Table 3.10 demonstrates that the means and standard deviations of RMSEs from 50 prediction results. As can be seen in Table 3.10, the prediction performance of the CGMY model has a very large variance compared to GBNN, and from Figure 3.7 we can also see that the RMSE is very large at several iterations. This suggest that the CGMY model often reaches at the local solution during the calibration process, resulting in parameter estimation that is not suitable for the prediction. This result supports that the GBNN is a more consistent method for price prediction than the pricing methods under the CGMY model.

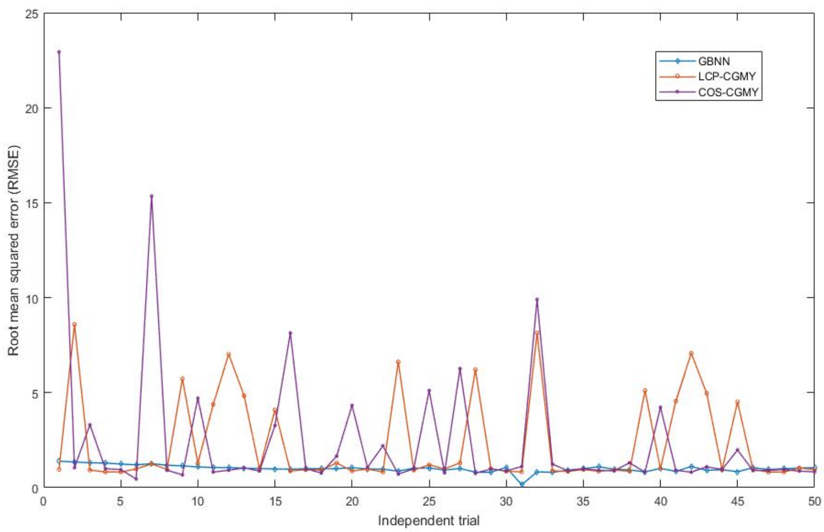


Figure 3.7 Fifty independent RMSE performance of each models.

3.5 Chapter Summary

Financial modeling is a matter of grave concern in the financial industry, and many researchers have struggled to elucidate complicate financial markets by proposing classical option models (e.g., Barone-Adesi Whaley methods, LSM methods, and ad-hoc methods) and machine learning models (e.g., artificial NNs, support vector machines, and GPs).

In this chapter, we proposed a generative model sampler based on variational inference to overcome the calculation limitation of explicit posterior probability density by generating a sample from the posterior distribution. Then, the validity of each model is investigated to elucidate the structure of option markets by comparing the performance of the models in terms of model calibration, prediction, and domain adaptation using the S&P 100 American/European put options.

First, the econometric jump models in model calibration using only the information of the previous day exhibited valid calibration results similar with those of the best-performing machine learning models, which used considerable information from the previous seven days. Second, econometric jump models for the model prediction of the one day and seven days ahead exhibited better performance than machine learning models. The price forecasts of the former for the next day or seven days were stable, whereas the latter decreased rapidly with the increase of prediction period. The robustness of the calibrated parameters for the former relative to the calibrated weights of the latter implied that the return processes of econometric jump models are stable over some periods and validated the better prediction results of the former than the latter. Finally, econometric jump models displayed successful domain adaptation performance, whereas the machine learning models did not. The latter failed to recognize the difference between American and European options and could not satisfactorily improve prediction accuracy regardless of adopted domain adaptation techniques for machine learning. From the empirical study, we concluded that econometric jump models can exhibit better performance of model estimation, prediction, and domain adaptation than machine learning models given the same information, such as expiration date and strike prices of contingent claims. Hence, machine learning models should integrate prior knowledge, such as no-arbitrage conditions, to avoid price distortions and to increase predictability. They should also develop a mechanism for generating the price process explicitly to improve domain adaptability, which we will study in the next chapter.

Based on the empirical result, we suggested the use of the generative Bayesian neural networks incorporating prior consistent with no-arbitrage pricing structure. First, in the model calibration, the classical American option models showed slightly better calibration results than the machine learning models and particularly noticeable differences during the financial crisis periods. The latter showed more sensitive reaction to the unusual market situations than the former. Second, in the model prediction of 1 day ahead, most machine learning models showed overall better performance than the classical American option models. Especially, a generative Bayesian neural network model showed the best overall performance. For the prediction results of 7-day ahead, the generative Bayesian neural network is superior to the CGMY model. It suggests that GBNN's robustness to the long-time window of the learned weight. This supports the need to use a prior information incorporating some financial market structures such as no-arbitrage constraints to the learning models.

Overall, we conclude that machine learning models can obtain a quantitative representation of option pricing more effectively than classical American option models given the same information, such as expiration date and strike prices of contingent claims. One can take advantage of machine learning models in the financial sector by incorporating more information and input variables, such as documents or sentiments, which needs to be further investigated.

Chapter 4

Predictive Models for Blockchain and Cryptocurrency Market

4.1 Chapter Overview

Bitcoin is a successful cipher currency introduced into the financial market based on its unique protocol and Nakamoto's systematic structural specification (Nakamoto, 2008). Unlike existing fiat currencies with central banks, Bitcoin aims to achieve complete decentralization. Participants in the Bitcoin market build trust relationships through the formation of Blockchain based on cryptography techniques using hash functions. Inherent characteristics of Bitcoin derived from Blockchain technologies have led to diverse research interests not only in the field of economics but also in cryptography and machine learning.

In this chapter, we train a Bayesian neural network based on the blockchain and prices data for predicting the Bitcoin process and try to account for the recent stochastic process shown in Figure 4.1, which has not been considered in previous studies. A BNN includes a regularization term into the objective function to prevent the overfitting problem that can be crucial to our framework. When the machine considers a lot of input variables, a trained machine can be complex and suffer from the overfitting problem. BNN models showed their effect to the financial derivative securities analysis (Gençay & Qi, 2001a). Formation of Blockchain, a core technology of Bitcoin, distinguishes Bitcoin from other fiat currencies and is directly related to Bitcoin's supply and demand. To

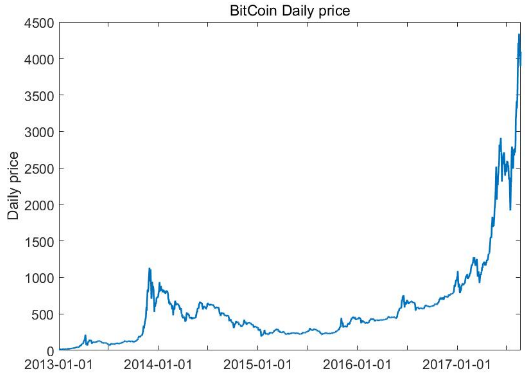


Figure 4.1 Bitcoin daily price(USD), from Sep-11 2011 to Aug-22 2017

the best of our knowledge, in addition to macroeconomic variables, direct use of Blockchain information, such as hash rate, difficulties, and block generation rate, has not been investigated to describe the process of Bitcoin price. To fill this gap, the current study systematically evaluates and characterizes the process of Bitcoin price by modeling and predicting Bitcoin prices using Blockchain information and macroeconomic factors.

After the bitcoin paved the way for the peer-to-peer decentralized cryptocurrency, several alternative cryptocurrencies are proposed to cope with perceived limitations of the bitcoin. They generally aim to the peer-to-peer and decentralization properties similar to the bitcoin and can be implemented via the blockchain or through other forms such as a directed acyclic graph. They can be launched by the forking in the existing cryptocurrency such as the bitcoin cash, bitcoin gold and the ethereum classic. Because these cryptocurrencies have essentially similar aspects, the analysis of the relationship between cryptocurrencies can results in the valuable meaning and can be applied to the clustering of the cryptocurrencies. Therefore, we proposed the enhanced GRU model based on the VAR model to analyze and visualize the relationship between cryptocurrencies.

4.2 Economics of Bitcoin and Blockchain

Barro's model (Barro, 1979) provides a simple Bitcoin pricing model under perfect market conditions as in (Ciaian et al., 2016). In this model, Bitcoin is assumed to possess currency value and is exchangeable with traditional currencies, which are under central bank control and can be used for purchasing goods and services. The total Bitcoin supply, S_B, is represented by

$$S_B = P_B B \tag{4.1}$$

where P_B denotes the exchange rate between Bitcoin and dollar (i.e. dollar per unit of Bitcoin), and B is the total capacity of Bitcoins in circulation.

The total Bitcoin demand depends on the general price level of goods or services, P; the economy size of Bitcoin, E; and the velocity of Bitcoin, V, which is the frequency at which a unit of Bitcoin is used for purchasing goods or services. The total demand of Bitcoin,D_B, is described as followed by:

$$D_B = \frac{PE}{V} \tag{4.2}$$

The market equilibrium with the perfect market assumption is acquired when the supply and the demand of Bitcoin is the same amount. The equilibrium is therefore achieved at

$$P_B = \frac{PE}{VB} \tag{4.3}$$

This equilibrium equation implies that in the perfect market, the Bitcoin price in dollars is affected proportionally by the general price level of goods or services multiplied by the economy size of Bitcoin, and inversely by the velocity of Bitcoin multiplied by the capacity of the Bitcoin market. The general price level of goods or services, P, can be determined indirectly from the global macroeconomic indexes in actual markets. The exchange rate between several fiat currencies and Bitcoin price describes the relationship between actual markets and Bitcoin market. The main difference between the Bitcoin market and general currency markets originates from the fact that the Bitcoin is a "virtual currency based on Blockchain technologies". Therefore, economic size, E; the velocity, V; and the capacity of the Bitcoin market, B, are closely related with several measurable market variables extracted from the Blockchain platform and, which will be reviewed in the next subsection.

Decentralization is the value pursued by all cryptocurrencies as opposed to general fiat currencies being valued by central banks. Decentralization can be specified by the following goals: (i) Who will maintain and manage the transaction ledger? (ii) Who will have the right to validate transactions? (iii) Who will create new Bitcoins? The blockchain is the only available technology that can simultaneously achieve these three goals. Generation of blocks in the Blockchain, which is directly involved in the creation and trading of Bitcoins, directly influence the supply and demand of Bitcoins. Combination of Blockchain technologies and the Bitcoin market is a real-world example of a combination of high-level cryptography and market economies.

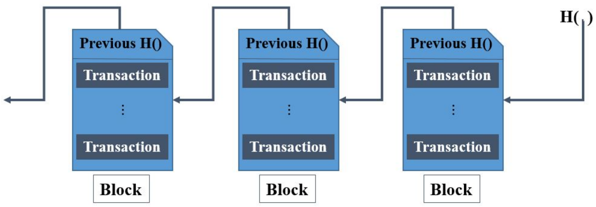


Figure 4.2 The formation of the Blockchain

We then describe in detail how the Blockchain can achieve the abovementioned goals in Bitcoin environment (Narayanan et al., 2016). A participant in a Bitcoin network acts as a part of a network system by providing hardware resources of their own computer, which is called a "distributed system". All issuance and transaction of money are conducted through P2P networks. All trading history is recorded in the Blockchain and shared by the network, and all past transaction history is verified by all network participants. The unit called "block", which includes recent transactions and a hash value from the previous "block", creates irreversible data by a hash function, and is pointed out from the next block. Figure 4.2 shows the general structure of Blockchain. It takes more than a certain amount of time to generate the block to make impossible to forge all or part of the Blockchain. This algorithm is called proof of work (PoW), and the difficulty is automatically set to ensure that the problem can be solved within approximately 10 minutes. PoW also provides incentives to motivate participants to maintain the value of Bitcoin by paying Bitcoin for the participant who created the block.

PoW agreement algorithm comes with several inherent risks. First, the validity of the block can be intervened when the majority of total participants is occupied by a group with a specific purpose called 51% problem. Second, when the Blockchain is forked, a considerable amount of time is consumed to form the agreed Blockchain until the longest chain is selected after generation of several blocks. This condition causes a transaction delay because the transaction cannot be completed during that time. Lastly, there may be the capacity limit of the Blockchain or the performance limit of each node. Safety of the current Blockchain can be monitored by observing measurable variables in the Blockchain from https://blockchain.info/.

Considering that supply and demand of Bitcoin are affected directly or indirectly by measurable variables involved in the formation of a Blockchain, the current study evaluates several variables related to Blockchain formation as features of the Bitcoin pricing process. Section IV describes in detail the variables exploited in empirical experiments.

4.3 An Empirical Study on Modeling and Prediction of Bitcoin Prices Based on Blockchain Information

4.3.1 Data Specification and Structure of the Experiment

Figure 4.1 shows the time series of Bitcoin price obtained from https:// bitcoincharts.com/markets/, where the value of 1-Bitcoin, which was about \$ 5 in September 2011, approximates \$ 4,000 in August 2017. During this period, market volatility with enormous price changes in Bitcoin becomes exceptional compared with that in traditional currency markets. It is evident that standard economic theories are insufficient to account for the impressive price development and volatility of Bitcoin (Kristoufek, 2013). Bitcoin markets do not possess purchasing power nor interest rate parity. In particular, Bitcoin is an actual implementation of decentralization issued under the consent of participants and not the central bank. This fact suggests that the need for completely new determinants of Bitcoin price: *the Blockchain information* that includes relevant features as main determinants for pricing Bitcoin. Blockchain data used for empirical analysis can be collected from https://blockchain.info/. Table 4.1 presents the Blockchain data and macroeconomic variables to be used in predicting the evolution of Bitcoin prices.

Table 4.1 Data for the empirical study

Data category	**Data**
Response var.	prices or log prices of Bitcoin(USD), vol. or log vol. of Bitcoin(USD)
Blockchain information	Trading vol.(USD,CNY), avg. block size, transactions/block, median confirm. time, hash rate, difficulty, cost % of trans., miners' rev., confirmed trans., total num. of uniq. Bitcoin
Macro economic development	S& P500, Eurostoxx, DOW30, NASDAQ, Crude oil, SSE, Gold, VIX, Nikkei225, FTSE100
Global currency ratio($\cdot$/USD)	GBP, JPY, CHF, CNY, EUR

Several blockchain variables are considered as follow:

? **Average block size (MB)**: the size of a block verified by all participants.

? **Transactions per block**: average number of transactions per block.

? **Median confirmation time**: the median time for each transaction to be accepted into a mined block and recorded to the ledger.

? **Hash rate**: estimated number of Tera (trillion) hashes per a second all miners (market participants to solve a hash problem for making a block) is performing.

? **Difficulty**: next difficulty =(previous difficulty $*2016*10$ minutes)/(time to mine last 2016 blocks)

? **Cost % of a transaction**: miners' revenue as the percentage of the transaction volume.

? **Miners revenue**: Total value of coin-base block rewards and transaction fees paid to miners.

? **Confirmed transaction**: the number of confirmed the validity of transactions per day.

? **Total number of a unique Bitcoin**: market capitalization of Bitcoin.

By employing ordinary least square (OLS) estimation, (van Wijk, 2013) demonstrates that the Dow Jones index, the euro-dollar exchange rate, and WTI oil price influence the value of Bitcoin price in the long run. We also consider several variables such as S& P500, Eurostoxx, DOW30, NASDAQ, Crude oil, SSE, Gold, VIX, Nikkei225, and FTSE100, which associated with global macroeconomic development.

Given that Bitcoin is related to traditional currency markets in addition to the cryptocurrency market itself based on digital cryptography, we take into account the exchange rates between global monetary markets; exchange rates are basic factors in the analysis of traditional currency markets. We specifically use *exchange rates between major fiat currencies* (GBP, JPY, CHF, CNY, EUR) and the dollar because these rates are most likely to affect the Bitcoin price.

In summary, we cover the daily data from Sep 11, 2011, to Aug 22, 2017 in the empirical analysis by employing both the traditional determinants of currency markets, such as global macro-economic development and the features endowed from the cryptocurrency. This experiment, which has not been performed in previous studies, primarily aims to discover the main features that can explain the recent highly volatile Bitcoin process.

Table 4.2: Summary statistics of the data

Data category	**Whole range**		**Recent 2 years**	
	mean	**stdev.**	**mean**	**stdev.**
Bitcoin price (USD)	458.32	606.2	901.96	804.0
log of Bitcoin price (USD)	5.04	1.92	6.52	0.71
Volatility	10.75	25.06	21.83	38.88
Trading volatility (BTC)	$6.66*10^4$	$5.82*10^4$	$7.15*10^4$	$5.21*10^4$
Trading volatility (USD)	$3.36*10^7$	$6.59*10^7$	$6.96*10^7$	$9.77*10^7$
Average block size	$3.94*10^5$	$3.21*10^5$	$7.84*10^5$	$1.65*10^5$
Transactions per block	751.81	625.03	1507.61	389.58
Median confirmation time	9.15	3.59	10.21	3.44
Hash rate	$8.14*10^6$	$1.41*10^6$	$2.18*10^6$	$1.68*10^6$
Difficulty	$1.08*10^{11}$	$1.86*10^{11}$	$2.9*10^{11}$	$2.21*10^{11}$
Miners revenue (%)	2.7	2.17	1.04	0.42
Miners revenue (USD)	$1.36*10^6$	$1.38*10^6$	$2.16*10^6$	$1.57*10^6$
Confirmed transac. per day	$1.14*10^5$	$9.29*10^4$	$2.26*10^5$	$5.83*10^4$
S&P 500	1851.29	346.26	2169.8	166.84
Eurostoxx	2977.97	413.73	3208.97	235.1
Dow Jones 30	$1.64*10^4$	$2.59*10^3$	$1.87*10^4$	$1.71*10^3$
Nasdaq	4279.47	1029.08	5289.29	543.53

Table 4.2: Summary statistics of the data

Data category	Whole range		Recent 2 years	
	mean	**stdev.**	**mean**	**stdev.**
Crudeoil	73.53	25.21	45.23	5.98
SSE	2706.43	633.03	3140.39	223.83
Gold	1356.01	201.03	1218.72	78.67
VIX	16.02	5.09	15.12	4.65
Nikkei225	$1.50*10^4$	$3.87*10^3$	$1.82*10^4$	$1.46*10^3$
FTSE100	6444.86	549.54	6704.23	531.92
USD/CNY	6.37	0.25	6.65	0.2
USD/GBP	0.67	0.06	0.74	0.06
USD/JPY	102.36	14.67	112.4	6.44
USD/EUR	0.82	0.08	0.91	0.03
USD/CHF	0.95	0.04	0.99	0.02

Table 4.3.1 shows summary statistics of response variables, Blockchain-related variables, global macroeconomic indexes, and international exchange rates used in empirical analysis from September 13, 2011, to July 21, 2017. Several notable points are considered in the empirical analysis. As shown in Table 4.3.1, response variables and Blockchain related variables in the last two years are considerably more variable than other categories such as global macroeconomic indexes and international exchange rates. Bitcoin prices and volatilities have nearly doubled over the past two years. In addition, Blockchain data exhibit a significant increase in trading volume and size per a block and a huge reduction in miner's profit and the hash rate.

On the other hand, there is little difference between the most recent two years and the overall range in the volatility of the global exchange rate market as well as the growth of the global macroeconomic market economy over the past two years is much smaller than that of Bitcoin. These results provide empirical evidence for the fact that the recent volatility in Bitcoin prices stems mostly from the Blockchain information directly involved in supply and demand of Bitcoin and not from other macro-financial markets.

Most of the previous studies have focused on either modeling Bitcoin price without considering its relationship to Blockchain information or identifying only its "linear" relationship to macroeconomic factors. The present study attempts to overcome these limitations by employing a Bayesian NN model that can investigate nonlinear influences of each relevant feature of input variables, the Blockchain information, and macroeconomic factors, on Bitcoin price formation. To this end, we first train a Bayesian NN to model Bitcoin price formation using given above-mentioned relevant features of the process. We have evaluated Bayesian NN in terms of training and test errors by using the representative non-linear methodologies, SVR, and the linear regression model as the benchmark methods.

Next, we develop a prediction model of the near-future price of Bitcoin after modeling the entire process. We configure forecasting models by *the rollover framework*, which is generally applied to portfolio theory. Rollover strategy is known as rolling a position forward which is closing out an old position and establishing a new position in a contract of the portfolio with a long time to maturity. In our experiments, the trained machine is closing out an old information and acquiring new data according to the rollover framework over time. Figure 4.3 shows a schematic rollover strategy employed in our empirical studies. At the initial training step, the machine is learned with N_{train} training data, and the prediction performance is measured using N_{test} test data. Next, after $t' - t$ time from time t, the machine is trained using again the N_{train} data from time t' to update old learning data, and the performance of N_{test} test data is thereafter measured. The machine is trained through the entire range in this way, and the average performance of prediction errors measured several times is evaluated.

Learning the machine through the rollover framework aims to validate the method of forecasting the next order of N_{test} test data from N_{train} training

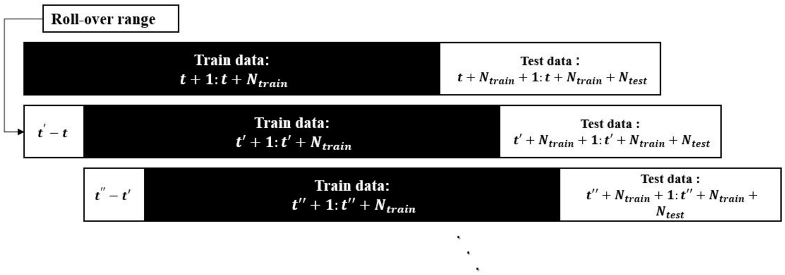


Figure 4.3 the formation of the Blockchain

data. Given that the model employs time series in batch format, it is is faster and easier to learn than other sequential neural networks models, LSTM or RNN, and can reflect the flow of information that changes with time. The rollover framework can be used to implement semi-online prediction models to incorporate new information or shocks with short learning time.

4.3.2 Linear Regression Analysis

We first construct a linear model for analysis of Bitcoin price and address several critical issues in assumptions of the linear regression model. A basic assumption required for linear regression is *the model assumption* that linear relationships exist between response variables and independent variables (Gujarati & Porter, 1999). Table 4.3.2 shows (linear) correlations between explanatory variables and response variables. Each column represents linear correlation coefficients of regressors for each response variable and the value in parentheses represents the results of t-test for the null hypothesis that there is no linear relationship between the two variables. We denote the null hypothesis-rejecting variables as bold, based on a p-value of 0.05, and presented a t-value because the p-value was as small as zero. We exclude the return as response variable because almost all values of correlation coefficients of each explanatory variable are not exceptionally significant for the return value of Bitcoin.

Table 4.3: Correlation coefficients and (t-values) between the response and independent variables.

Data category	return	price	log(price)	log(vol.)
Trading volatility (BTC)	**0.064** **(2.987)**	**0.071** **(3.315)**	**0.123** **(5.772)**	**0.245** **(11.769)**
Trading volatility (USD)	0.016 (0.745)	**0.777** **(57.485)**	**0.474** **(25.071)**	**0.683** **(43.549)**
Average block size	0.001 (0.047)	**0.663** **(41.246)**	**0.744** **(51.857)**	**0.404** **(20.569)**
Transactions per block	0.011 (0.512)	**0.647** **(39.518)**	**0.715** **(47.63)**	**0.39** **(19.725)**
Median confirmation time	0.04 (1.864)	**0.26** **(12.54)**	0.018 (0.838)	**0.163** **(7.694)**
Hash rate	0.025 (1.165)	**0.9** **(96.16)**	**0.577** **(32.902)**	**0.583** **(33.419)**
Difficulty	0.024 (1.118)	**0.906** **(99.686)**	**0.58** **(33.159)**	**0.588** **(33.856)**
Miners revenue (%)	-0.034 (-1.584)	**-0.34** **(-16.838)**	**-0.51** **(-27.613)**	**-0.24** **(-11.514)**
Miners revenue (USD)	-0.015 (-0.699)	**0.92** **(109.326)**	**0.76** **(54.46)**	**0.625** **(37.288)**
Confirmed trans. per day	0.008 (0.373)	**0.66** **(40.915)**	**0.731** **(49.891)**	**0.402** **(20.447)**
S&P 500	-0.006 (-0.279)	**0.691** **(44.52)**	**0.928** **(116)**	**0.415** **(21.243)**
Eurostoxx	-0.008 (-0.373)	**0.537** **(29.647)**	**0.838** **(71.523)**	**0.339** **(16.782)**
Dow Jones 30	0.002 (0.093)	**0.746** **(52.171)**	**0.916** **(106.338)**	**0.454** **(23.731)**
Nasdaq	-0.007 (-0.326)	**0.722** **(48.599)**	**0.896** **(93.973)**	**0.442** **(22.948)**

Table 4.3: Correlation coefficients and (t-values) between the response and independent variables.

Data category	return	price	log(price)	log(vol.)
Crudeoil	0.015 (0.699)	**-0.401** **(-20.386)**	**-0.545** **(-30.273)**	**-0.264** **(-12.747)**
SSE	-0.018 (-0.838)	**0.27** **(13.06)**	**0.408** **(20.813)**	**0.184** **(8.718)**
VIX	-0.051 (-2.378)	**-0.384** **(-19.369)**	**-0.544** **(-30.194)**	**-0.215** **(-10.253)**
Nikkei225	-0.011 (-0.512)	**0.553** **(30.911)**	**0.884** **(88.067)**	**0.346** **(17.175)**
FTSE100	0.016 (0.745)	**0.67** **(42.033)**	**0.843** **(72.987)**	**0.396** **(20.085)**
USD/CNY	0.013 (0.605)	**0.572** **(32.477)**	**0.355** **(17.685)**	**0.331** **(16.336)**
USD/GBP	0.019 (0.885)	**0.584** **(33.506)**	**0.477** **(25.276)**	**0.339** **(16.782)**
USD/JPY	-0.018 (-0.838)	**0.38** **(19.133)**	**0.819** **(66.475)**	**0.244** **(11.718)**
USD/EUR	-0.002 (-0.093)	**0.344** **(17.062)**	**0.496** **(26.603)**	**0.208** **(9.904)**
USD/CHF	0.008 (0.373)	**0.266** **(12.851)**	**0.341** **(16.894)**	**0.164** **(7.743)**
Gold	0.019 (0.885)	**-0.396** **(-20.085)**	**-0.858** **(-77.795)**	**-0.241** **(-11.565)**

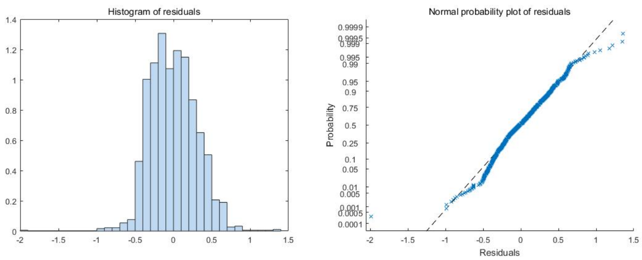
Next, we discuss the multicollinearity problem, which is often encountered in linear regression analysis. Several statistical problems are caused from the multicollinearity which is the situation that some regressors have a linear relationship with other regressors. It can cause undesirable regression analysis: very high R^2 for some coefficients that are not statistically significant and their t-statistics sensitive to data variation (Gujarati & Porter, 1999). One of the

Table 4.4 VIF values of each explanatory variable for detecting the collinearity proble

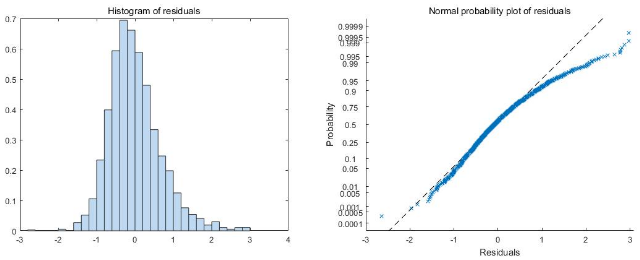
Data category	VIF	Data category	VIF	Data category	VIF
Trading vol. (BTC)	1.5688	**Trading vol. (USD)**	3.45327	**Avg. block size**	33.2689
Trans. per block	36.7642	**Median conf. time**	2.1306	**Hash rate**	122.3453
Difficulty	150.3203	**Miners revenue (%)**	2.4462	**Miners revenue (USD)**	8.2981
Confirmed trans. per day	48.1753	**S&P 500**	730.6197	**Eurostoxx**	41.9197
Dow Jones 30	402.9169	**Nasdaq**	304.5080	**Crudeoil**	22.8668
SSE	10.1965	**Gold**	21.4123	**VIX**	4.5702
Nikkei 225	128.2556	**FTSE100**	51.7874	**USD/CNY**	20.3706
USD/GBP	45.355	**USD/JPY**	58.1390	**USD/EUR**	43.6925
USD/CHF	7.7059				

prescriptions for dealing with multicollinearity is to do a linear regression except for variables with large VIF values, which is a sort of measure of the linear relationship between variables (Gujarati & Porter, 1999). To remove redundant variables for preventing the collinearity problems, we eliminate several explanatory variables with large VIF values. Table 4.4 shows VIF values of each explanatory variable. In this study, we have determined that the set of variables excluding linear relationships is suitable for linear regression analysis to avoid multicollinearity problem. We select 16 suitable discriminators after eliminating variables with large VIFs and perform linear regression analysis on Bitcoin log prices and log volatilities with these 16 discriminators. Removed variables include the following: transactions per a block, difficulty of the hash function, Nikkei225 index, S&P 500 index, Eurostoxx index, DOW30 index, NASDAQ, and exchange rates of EUR and GBP. From these 16 regressors, we construct two linear models, one for the log price and one for the volatility of Bitcoin process. We then evaluate assumption fitness, say the *residual assumption* that residual terms are independently and identically distributed.

Finally, we generate histograms residuals of each model to verify the residual assumption by confirming it follows a normal distribution.



(a) Histogram for log-price (b) QQ plot for log-price



(c) Histogram for log-volatility (d) QQ plot for log-volatility

Figure 4.4 Residual evaluations for (a) Histogram, (b) Normal probability (QQ) plot of the Bitcoin log price, and (c) Histogram, (d) Normal probability (QQ) plot of the Bitcoin log volatility

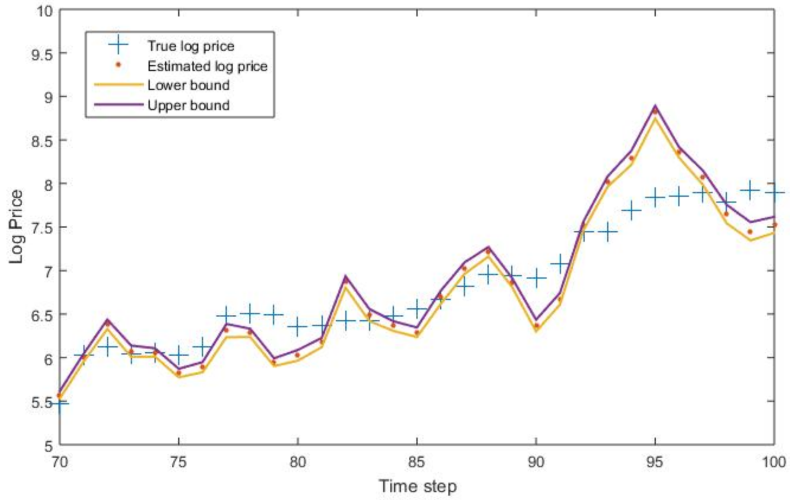
Figure 4.4 (a) & (b) show that the Bitcoin log price satisfies *the residual assumption* for linear regression: the histogram is bell-typed and symmetric and the QQ-plot shows a similar pattern with the normal distribution. By contrast, Figure 4.4 (c) & (d) show that residuals of the linear model for log volatility of Bitcoin do not follow a normal distribution with a positive-skewed histogram. Time series of log volatility of Bitcoin is therefore unsuitable for linear analysis except for the log price of Bitcoin due to the violation of each assumption.

Each linear model trained from a random 85% of whole data are disparate from true log prices or log volatilities. Figure 4.5 demonstrates that predicted log prices (volatilities) and a confidence interval of most recent 30 test data, implying the unsuitability of the linear model in predicting the time series of Bitcoin price. Figure 4.5 shows that most true values are out of the confidence interval of the linear model. This means that the learned linear model does not make an adequate prediction of the output value albeit in predicting trends in little.

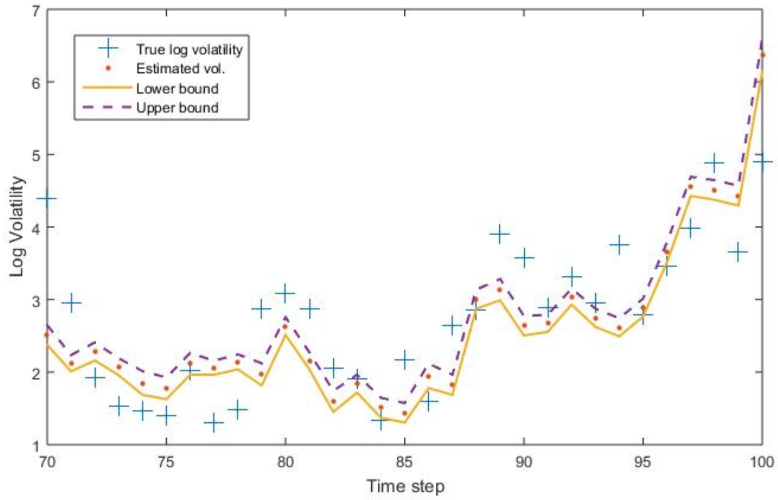
4.3.3 Estimation and Prediction Results of Bitcoin Price

We next perform time series analysis of Bitcoin prices using a BNN model and compare with the benchmark models, which are the linear regression and the SVR model. A total of 25 explanatory variables belonging to three categories are employed as inputs for BNN learning. We also address another input set that comprises 16 input variables by eliminating several unimportant variables as mentioned in the previous subsection. We consider two response variables, log price of Bitcoin and volatility of Bitcoin price, because extremely high volatility is an important feature of Bitcoin. In general, volatility is a significant variable assessed equally to the value of an option in economic analysis. We use log-scaled values of both output response variables to account for the large difference between Bitcoin value in the early period and its most recent value.

We train the BNN model through 10-fold cross-validation. To mitigate the effect of how to divide the data, we repeated hold-out validation steps where $\frac{9}{10}N$ training data and $\frac{1}{10}N$ test data, given the total number of the data is N. Where performances of each trained model are measured by root mean square



(a) Bitcoin log-price



(b) Bitcoin log-volatility

Figure 4.5 Prediction results of (a) the Bitcoin log price and (b) the Bitcoin log volatility

error (RMSE) and mean absolute percentage error (MAPE). Definitions of each evaluation criteria are as followings:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{4.4}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}| \tag{4.5}$$

where N is the number of samples, y_i is the i-th true objective value, and $\hat{y}_i$ is the i-th estimated value.

Table 4.5 Training error for the Bitcoin price formation

Response variable		**Log price**		**Log volatility**	
Number of Input variable		**26**	**16**	**25**	**16**
Linear	**RMSE**	-	0.0913	-	0.4595
Regression	**MAPE**	-	0.0681	-	0.5905
Bayesian	**RMSE**	0.0031	0.0047	0.1612	0.1717
NN	**MAPE**	0.0119	0.0148	0.3314	0.3512
Support vec.	**RMSE**	0.1453	0.1434	0.3810	0.3939
Regression	**MAPE**	0.0325	0.0322	0.5411	0.6293

Table 4.5 and 4.6 summarize results of training errors and test errors, respectively. We observe that BNN models outperform other models in terms of RMSE and MAPE for predicting the log price of Bitcoin. Log price of Bitcoin is learned exceptionally by the BNN model with training and test error of around 1% MAPE. In the case of log volatility, the prediction error of log volatility in the test phase is slightly larger than that in the training phase. BNN model is more reliable for describing the process of log volatility than other benchmark

Table 4.6 Test error for the Bitcoin price formation

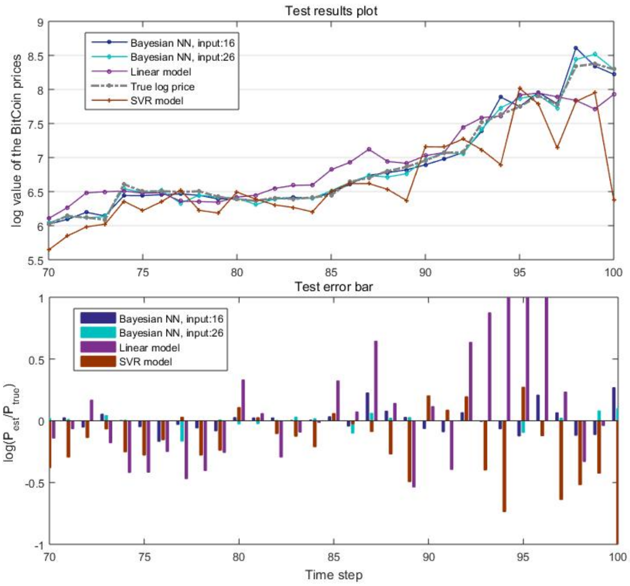
Response variable		Log price		Log volatility	
Number of Input variable		**26**	**16**	**25**	**16**
Linear	**RMSE**	-	0.0935	-	0.4823
Regression	**MAPE**	-	0.0712	-	0.6263
Bayesian	**RMSE**	0.0039	0.0069	0.2546	0.2325
NN	**MAPE**	0.0138	0.0180	0.5090	0.5222
Support vec.	**RMSE**	0.3201	0.2742	0.5487	0.5297
Regression	**MAPE**	0.0428	0.0404	0.7232	0.8629

models. After eliminating redundant variables from linear correlation analysis, the error value is relatively small when all 26 input variables are considered instead of the abridged 16 input variables. This condition implies that removed variables may explain nonlinear relationships to adequately account for response variables. SVR model shows poor performances in both training and test phase. From this results, we can confirm that Bayesian NN is better suited for the Bitcoin time series analysis than SVR albeit in they are included the same nonparametric model.

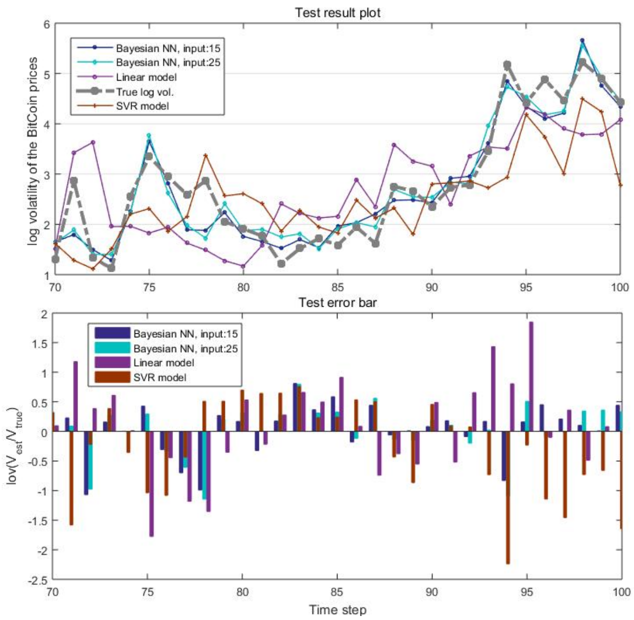
Figure 4.6 shows the values of estimated response variables for the recent 30 test input data according to time indexes. We observe that the recent volatile tendency is well expressed in terms of explanatory input variables. The case of log price presents a tendency for underestimation when price rises and overestimation when the price falls. In the case of the log price, we can see that all models predict the actual tendency of the price to some extent. On the other hands, in terms of error size, it is confirmed that other models are larger than that of Bayesian neural networks. There is no tendency of over- or underestimate in all models. Bayesian neural networks tended to predict consistent trends regardless of the number of inputs. In the case of volatility, the Bayesian NN model predicts better the direction of volatility than other benchmark models, and neither of the four models tends to over or under-estimate.

Finally, we provide prediction results of the trained BNN under the rollover framework. Rollover framework physically excludes old preceding data to reflect that the previous information shrinks as training is repeated. The construction method of the model in this subsection is fundamentally different from that of the previous subsection. In the previous subsection, we have extracted part of the entire data for training purpose, assuming that we have all data for the entire time range. Although the method in the previous section is adequate to assess how well the model has learned for the whole data, it is not appropriate to predict future outcome from the historical data.

We train the machine using data obtained 200 days before the present day and predict the current day's price from the trained machine under the rollover framework. Given that future data are not considered in the training phase, we can infer that prediction performance may be inferior to that of the previous subsection. Table 4.7 presents prediction error for Bitcoin price under the rollover framework. We note that overall performance is slightly poor compared with the model construction in the previous subsection. Nevertheless, prediction result for the log price of Bitcoin still maintains low error rates. By contrast, prediction errors are almost doubled for log volatility outputs. Figure 4.7 shows plots of prediction results for the log price and log volatility of Bitcoin. We show that log price is relatively well explained based on the employed input variables and during sudden fluctuations. In the case of log volatility, the discrepancy between true volatility and predicted volatility is relatively large, but directionality is well approximated. In summary, the learned BNN models can effectively describe the recent highly volatile Bitcoin price process and the price in the entire range.

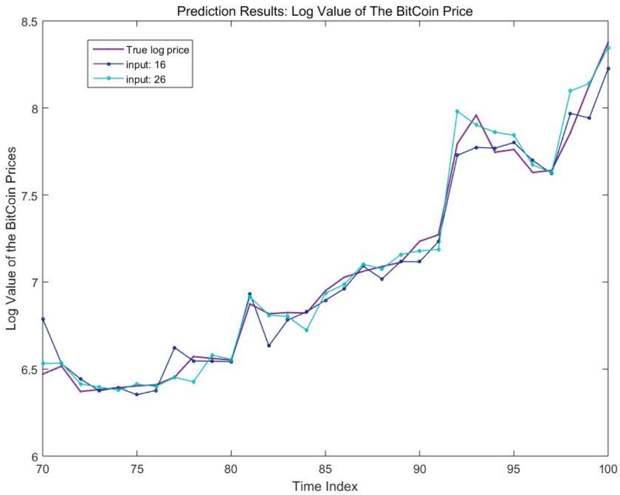


(a) Bitcoin log-price

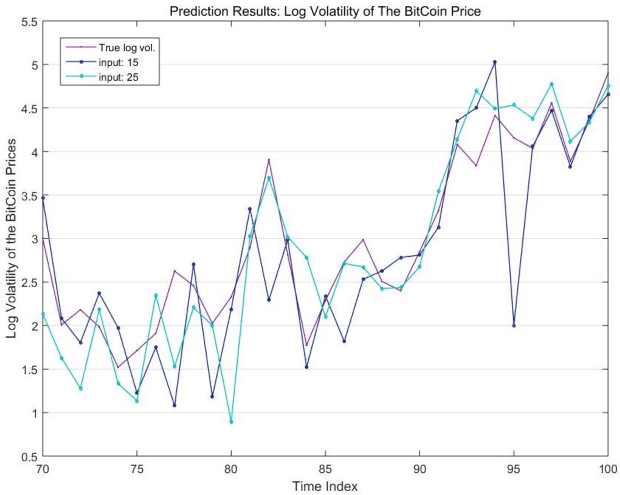


(b) Bitcoin log volatility

Figure 4.6 Test result plot of (a) the Bitcoin log price and (b) the Bitcoin log volatility



(a) Log value of the Bitcoin price



(b) Log volatility of the Bitcoin price

Figure 4.7 Prediction results of (a) the log value of the Bitcoin price and (b) the log volatility of the Bitcoin price.

Table 4.7 Prediction error for the Bitcoin price under the rollover framework

Response variable	Number of input variable	RMSE	MAPE
log	26	0.0256	0.0198
price	16	0.0244	0.0200
log	25	0.5750	0.8992
volatility	15	0.5114	0.6302

4.4 Enhanced GRU Framework for Correlation Analysis of Cryptocurrency Market

4.4.1 Enhanced GRU Framework

Recurrent Neural Network (RNN) is neural network model that is specialized for time series sequential data. Its network structure is similar to multilayer perceptron, but it contains previous hidden nodes as an inputs. Basic RNN models with model F, input x_t and hidden state h_t for time step t, is given as :

$$h_t = F(x_t, h_{t-1}, \theta) \tag{4.6}$$

Equation 4.6 is called recurrent because h_{t-1} goes recurrently back to input in the time t. Since h_{t-1} requires x_{t-1} as an input, this hidden state has the information of whole past sequence $(x_t, x_{t-1}, ..., x_2, x_1)$ as an input. Therefore RNN is widely used in the tasks that requires time sequential concepts in the model, such as neural machine translation(Cho et al., 2014) or speech recognition(Graves et al., 2013).

However deep RNN models has challenge of long term dependencies(LeCun et al., 2015). As RNN models gets deeper the model's gradients will be propagated over many times, and lead to gradient vanishing and exploding problem. To solve this problem, long short-term memory (LSTM) model was suggested by (Gers et al., 1999). The LSTM model uses the concept of gated RNN to solve long-term dependency problems and has been found very successfully in various applications. However predicting time-series Bitcoin data has different structure with other LSTM models, because whole dataset is composed as single time-series data. It is hard to construct validation set, because we can not directly apply validation methods such as cross-validation. In this dissertation, we suggested the enhanced GRU framework for the multivariate time series analysis. When we consider the following vector auto-regression (VAR)(1) model:

$$\mathbf{y_t} = \phi_{0t} + \mathbf{\Phi_t y_{t-1}} + \mathbf{a_t} \tag{4.7}$$

where $\mathbf{y}_t$ is the target n-dim variable at time t, ϕ_{0t} and $\mathbf{\Phi_t}$ are parameters for the VAR(1) model, and $\mathbf{a_t}$ is the shock at time t from the time invariant distribution with mean 0 and the covariance matrix σ. In the VAR(1) model, the parameter calibration is object to acquire suitable approximation of each ϕ_{0t}, $\mathbf{\Phi_t}$, and σ.The approximation can be achieved by the ordinary least square (OLS) methods under the assumptions that the covariance matrix is positive definite and the time series $\mathbf{y}_t$ is weakly stationary. However, the OLS estimation procedure is suffered from the above-mentioned assumptions and can be difficult to calculate when the dimension of the target vector is increased.

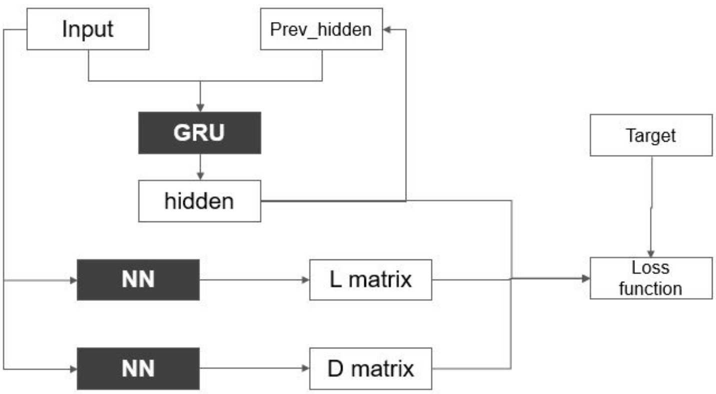


Figure 4.8 Enhanced GRU framework for multivariate time series analysis

To overcome the drawback, we suggested the enhanced GRU framework for the multivariate time series analysis. This model overcomes the difficulties of the existing VAR model estimation method by learning the components of the lower triangular matrix and the diagonal matrix from each independent neural network after performing the cholesky decomposition to satisfy the positive definite condition of the correlation matrix. Coefficients dependent on the time series vector was learned from the gate of GRU and tried to learn the change of time series vector over time. Figure 4.8 described the enhanced GRU framework for the multivariate time series analysis.

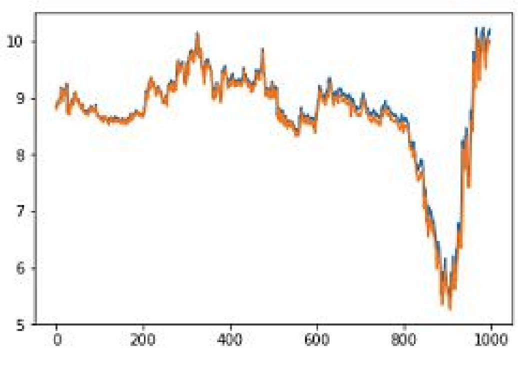
4.4.2 Empricial Studies

In this empirical study, we consider the 30-minute price data from November 17, 2015 to November 17, 2017 for a total of eight cryptographic currencies such as ETH, DASH, XRP, XMR, LTC, XEM, EMC2 and NXT.

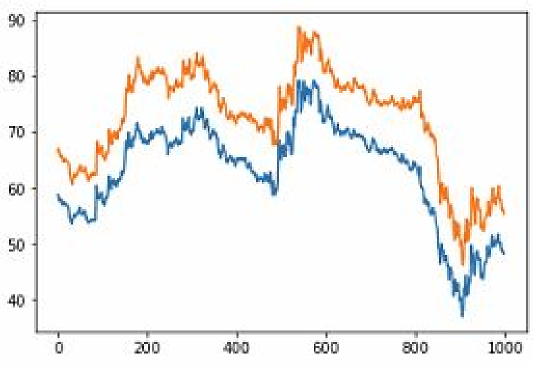
We train the model by using a total of about 38000 historical data and test the model by using the recent 2000 data. We considered the batch size is 45 days. It means that the GRU is trained with 45 batch data in a single iteration. We trained the enhanced GRU sequentially with altcoins as input in the order presented above. In other words, we use the 38000 training data of 45 batches to train the GRU and measure the test performance of the trained machine for the recent 2000 test data.

The following is an illustration of some of the predicted results. Figure 4.9a shows that the Litecoin is better approximated than Dash. From Figure 4.9b, it can be seen that the Dash is over-estimated at a constant rate, which does not directly extract the output from the GRU but looks like a gap in the process of performing a linear regression once more. It seems to be due to the difference in units of LTC and DASH.

Figure 4.10 shows that the correlation matrix between the entire cryptocurrencies at a particular point in time. We can confirm that the correlation matrix of shocks of cryptocurrencies price varies with time from the change of information of GRU gate. Figure 4.10 shows that there is a covariance close to zero



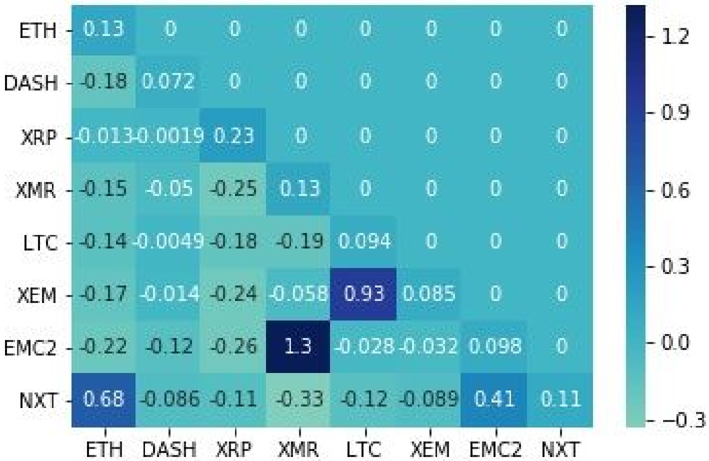
(a) Prediction results of the LTC



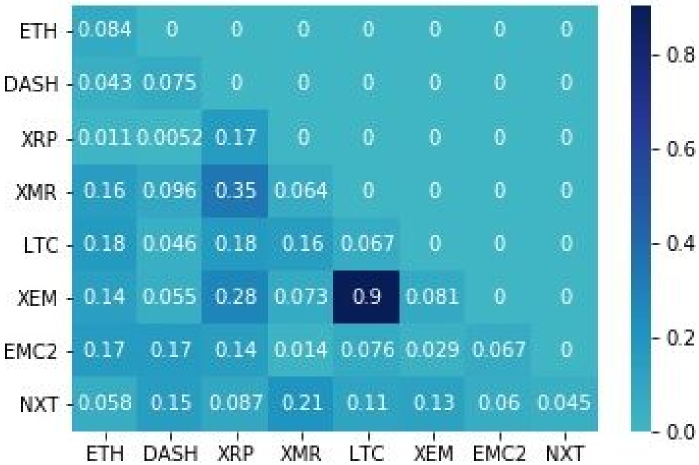
(b) Prediction results of the DASH

Figure 4.9 Prediction results of (a) Prediction results of the LTC and (b) Prediction results of the DASH. Blue line is true price and Orange line is estimated price

between variables at the beginning of learning. However, as the learning progresses, the value of the covariance matrix converges to the value corresponding to the covariance of each variable. The variance of each variable ranges from about 0.1 to 0.25, and the covariance matrix shows that ETH and XRP have opposite behavior to other currencies. In particular, EMC2 and NXT have large volatility with other currencies, which seems to be due to the low ratio of market capitalization.



(a) Covariance matrix result for 2017.09.12.



(b) Covariance matrix result (a) 2017.05.15.

Figure 4.10 Covariance matrix result (a) 2017.09.12 and (b) 2017.05.15.

4.5 Chapter Summary

Bitcoin is a successful cryptocurrency, and it has been extensively studied in fields of economics and computer science. In this study, we analyze the time series of Bitcoin price with a BNN using Blockchain information in addition to macroeconomic variables and address the recent highly volatile Bitcoin prices.

Given the data of the entire time range, experimental results show that the BNN model learned with the selected features effectively describes processes of Bitcoin log price and log volatility. Adoption of rollover framework experimentally demonstrates the predictive performance of BNN is better than other benchmark methods on log price and volatility processes of Bitcoin.

Through the empirical analysis, we have confirmed that the BNN model describes the fluctuation of Bitcoin up to August 2017, which is relatively recent. Unlike other benchmark models that fail directional prediction, the BNN model succeeded in relatively accurate direction prediction. From these experimental results, the BNN model is expected to have similar performance in more recent data. As the variation of Bitcoin process gets attention, it is expected that the expansion and application of the BNN model would be effective for the analysis and prediction of the Bitcoin process.

Investigating nonlinear relationships between input functions based on network analysis can explain analysis of Bitcoin price time series. Variability of Bitcoin must be modeled and predicted more appropriately. This goal can be achieved by adopting other extended machine learning methods or considering new input capabilities related to the variability of Bitcoin. Such study will contribute to rich Bitcoin time series analysis in addition to existing Bitcoin studies.

We have attempted to visualize the relationship between altcoins, in which the creation principles are closely related, unlike the existing stock market with independently issued stocks. To reveal the relationship, we have proposed the enhanced GRU framework based on the VAR model. In the enhanced GRU framework, the effect of each time series variable on each other is expressed as a linear regression coefficient through the VAR model consisting of a vector of time series of each alternative coin. In this case, each linear regression coefficient is dynamic parameters estimated through GRU and neural networks, unlike the original VAR model. From the proposed framework, we have confirmed that there is a significant correlation between altcoins by investigating empirical analysis based on eight alternative coins. The correlation analysis of cryptocurrencies is expected to contribute in part to the valuation of cryptographic currencies that have not yet been established.

Chapter 5

Conclusion

5.1 Contributions

This dissertation attempts to analyze the financial derivatives market and the cryptocurrency market using econometric models and machine learning models. Based on the results of systematic empirical experiments based on given market data, each models have been evaluated based on market explanatory ability and market prediction ability. We propose data-driven machine learning methodologies to improve the market predictability for each market.

When the hidden variables in the econometric models such as the GARCH model or the stochastic volatility model constitute a time series model separately from the observation variables, the existing parameter calibration methods causes slow convergence speed and frequent local solution problems. In particular, in the case of the general MCMC methodology where it is impossible to know the specific target probability distribution, it is an important factor to determine a candidate probability distribution close to or similar to the target probability distribution for the overall performance improvement. In this dissertation, we propose a MCMC framework that a large amount of samples is extracted from the candidate probability distribution nearest to the target probability distribution in terms of Kullback-Leibler (KL) distance by using the generative model concept. It is possible to extract the samples in a very short time, since the sample is acquired from the generative model. We have improved the disadvantages of the existing MCMC methodology, which is dependent on the choice of the candidate probability distribution, by providing only samples from the nearest probability distribution in terms of KL divergence instead of suggesting a specific probability distribution.

Given S&P index option data in 2012, in-sample and out-of-sample are measured to compare model validity and predictability of the representative econometric model, the CGMY and Kou model, and conventional machine learning models such as artificial neural networks, Bayesian artificial neural networks, support vector machines, and Gaussian process models. In the case of model estimation, the jump diffusion models have the best performance in estimating the model using the data of the recent day, whereas the machine learning model has the highest model estimation performance using the data of the last week. On the other hand, the performance of the jump diffusion models and the Bayesian artificial neural network were the best in forecast, and the performance of the other machine learning decreased rapidly as the range of prediction increased. Especially, it was confirmed that the jump diffusion model has a very high performance in terms of domain adaptation between the American option and the European option. This difference is reflected in the fact that the jump diffusion model is based on the common asset of the American option and the European option.

Based on this empirical precedent study, we proposed a machine learning model called generative Bayesian neural network (GBNN) to overcome the disadvantages of the machine learning model. Since the general machine learning methodology learns the model from the data, the performance decreases very rapidly in the deep ITM or deep OTM domain with few data point when learning the model for the option market. During the initial learning, GBNN acquired an appropriate price in any area by adding virtual price data from an arbitrarary jump-diffusion model. When the next learning period comes, GBNN maximizes posterior probability through the GBNN obtains pror information from the GBNN data learned up to the previous day, and learns likelihood probability from actual trading data of learning day. As a result, GBNN's deep ITM and

OTM areas have significantly improved estimation and prediction performance and are much better than the jump-diffusion models unlike other machine learning models for the S & P 100 index American option data from 2003 to 2012. In particular, in the previous study, GBNN showed that the model estimation performance was very fast and stable compared to other methodologies, unlike the general machine learning model, where the model fit performance is highly volatile according to the applied data range in terms of model estimation. In addition, we can confirm that the GBNN is much faster in terms of option price calculation time for the fitted model. Econometric models calculate the option price after given the asset value based on the obtained parameter even after the model is formed. It reflect the characteristics of the artificial neural network which shows very fast speed in the test side after the model is formed.

In recent years, a variety of cryptographic currencies have been developed, beginning with the first cryptocurrency Bitcoin, which technically implemented the concept of distributed ledger proposed by Satoshi Nakamoto in 2008. There is a growing demand for new analytical technology as well as traditional market analysis techniques for the cryptocurrency market, which has new and unique features that have not been existing. In this dissertation, we use quantitative data of Blockchain technically implemented decentralized branch to analyze the representative cryptocurrency Bitcoin time series studied by previous literature based on conventional methodology of econometrics. Bayesian neural networks considering block-chain data show higher predictive performance and estimation performance than other benchmark models, and identify the recent volatility of cryptography compared to previous studies. Correlation analysis between cryptocurrencies is performed using the enhanced GRU model framework, under the assumption that there will be a correlation between the prices of cryptographic currencies since many altcoins derived from Bitcoin are technically developed from the same root code. The vector autoregressive (VAR) model, which is a traditional market model, is based on the assumption that the correlation between the variables is a linear model. Assuming that the gate value obtained from the GRU model is the parameter of the VAR model, The covariance matrices of the cryptosystem are estimated through the artificial neural network, which makes it possible to visualize the correlation between various alternative currencies in the cryptographic market. As a result, it is confirmed that there is a close correlation between alternative currencies. Especially, it is confirmed that there is a very significant correlation between the currencies separated from the existing currencies and the existing currencies.

This dissertation has developed data-driven technologies for the time series analysis of derivatives and cryptocurrency market, and conducted quantitative analysis of the market. There has been limited research on the machine learning framework of interpretable data based. This dissertation focuses on visualizing and interpreting meaning from the data by developing a machine learning based model easy to interpret. This dissertation can contribute to the analysis of time series of recently formed cryptocurrency market. In addition, it is expected that the application of the time series analysis framework of derivatives based on the data can be applied to expand to the analysis of derivatives market with the cryptocurrency underlying.

5.2 Future Work

Several limitation of the dissertation should be addressed in future work. First, the econometric model for calculating the prior virtual prices used in the proposed generative Bayesian neural network model should be updated with time. It may cause that the GBNN can learn the wrong prior information if the outdated econometric model does not reflect the current market information after time. In this dissertation, we consider the prior knowledge obtained only from the jump-diffusion model. The extended research topic can be considered the GBNN when acquiring the prior information from the model considering the variation of the volatility such as the stochastic volatility model or the GARCH model was selected. In cryptocurrency market analysis, this study has the limitation that only the quantitative data constituting the blockchain such as the difficulty and the hash rate are considered. For a rich and systematic analysis of the money market, fundamental studies on the basic mechanisms of blockchain technology and the analysis of the value of cryptography should be accompanied.

Bibliography

AitSahlia, F., Goswami, M., & Guha, S. (2010). American option pricing under stochastic volatility: an empirical evaluation. *Computational Management Science*, *7*(2), 189–206.

Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 269–342.

Bakshi, G., Cao, C., & Chen, Z. (1997). Empirical performance of alternative option pricing models. *The Journal of Finance*, *52*(5), 2003–2049.

Bariviera, A. F., Basgall, M. J., Hasperué, W., & Naiouf, M. (2017). Some stylized facts of the bitcoin market. *Physica A: Statistical Mechanics and its Applications*, *484*, 82–90.

Barndorff-Nielsen, O. E. (1997). Processes of normal inverse gaussian type. *Finance and stochastics*, *2*(1), 41–68.

BARONE-ADESI, G., & Whaley, R. E. (1987). Efficient analytic approximation of american option values. *The Journal of Finance*, *42*(2), 301–320.

Barro, R. J. (1979). Money and the price level under the gold standard. *The*

Economic Journal, *89*(353), 13–33.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, *79*(1), 151–175.

Benko, M., Fengler, M., Härdle, W., & Kopa, M. (2007). On extracting information implied in options. *Computational statistics*, *22*(4), 543–553.

Berkowitz, J., et al. (2010). On justifications for the ad hoc black-scholes method of option pricing. *Studies in Nonlinear Dynamics and Econometrics*, *14*(1), 1–27.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of political economy*, *81*(3), 637–654.

Blair, B. J., Poon, S.-H., & Taylor, S. J. (2010). Forecasting s&p 100 volatility: the incremental information content of implied volatilities and high-frequency index returns. In *Handbook of quantitative finance and risk management* (pp. 1333–1344). Springer.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, *31*(3), 307–327.

Bouri, E., Molnár, P., Azzi, G., Roubaud, D., & Hagfors, L. I. (2017). On the hedge and safe haven properties of bitcoin: Is it really more than a diversifier? *Finance Research Letters*, *20*, 192–198.

Brandt, M. W., & Wu, T. (2002). Cross-sectional tests of deterministic volatility functions. *Journal of Empirical Finance*, *9*(5), 525–550.

Burden, F., & Winkler, D. (2009). Bayesian regularization of neural networks. *Artificial Neural Networks: Methods and Applications*, 23–42.

Carr, P., Geman, H., Madan, D. B., & Yor, M. (2003). Stochastic volatility for lévy processes. *Mathematical Finance*, *13*(3), 345–382.

Carter, C. K., & Kohn, R. (1994). On gibbs sampling for state space models. *Biometrika*, *81*(3), 541–553.

Cheah, E.-T., & Fry, J. (2015). Speculative bubbles in bitcoin markets? an empirical investigation into the fundamental value of bitcoin. *Economics Letters*, *130*, 32–36.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, *89*(3), 539–552.

Chopin, N., Jacob, P. E., & Papaspiliopoulos, O. (2013). Smc2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *75*(3), 397–426.

Christoffersen, P., & Jacobs, K. (2004). Which garch model for option valuation? *Management science*, *50*(9), 1204–1221.

Chu, J., Nadarajah, S., & Chan, S. (2015). Statistical analysis of the exchange rate of bitcoin. *PloS one*, *10*(7), e0133678.

Ciaian, P., Rajcaniova, M., & Kancs, d. (2016). The economics of bitcoin price formation. *Applied Economics*, *48*(19), 1799–1815.

Clément, E., Lamberton, D., & Protter, P. (2002). An analysis of a least squares regression method for american option pricing. *Finance and Stochastics*, *6*(4), 449–471.

Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, *12*(656-704), 3.

Dumas, B., Fleming, J., & Whaley, R. E. (1998). Implied volatility functions: Empirical tests. *The Journal of Finance*, *53*(6), 2059–2106.

Dupire, B. (1997). Pricing and hedging with smiles. *Mathematics of derivative securities*, *1*(1), 103–111.

Durbin, J., & Koopman, S. J. (2002). A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, *89*(3), 603–616.

Dyhrberg, A. H. (2016a). Bitcoin, gold and the dollar–a garch volatility analysis. *Finance Research Letters*, *16*, 85–92.

Dyhrberg, A. H. (2016b). Hedging capabilities of bitcoin. is it the virtual gold? *Finance Research Letters*, *16*, 139–144.

Eberlein, E., Keller, U., et al. (1995). Hyperbolic distributions in finance. *Bernoulli*, *1*(3), 281–299.

Ederington, L. H., & Guan, W. (2002). Is implied volatility an informationally efficient and effective predictor of future volatility?

Fang, F., & Oosterlee, C. W. (2008). A novel pricing method for european options based on fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, *31*(2), 826–848.

Fang, F., & Oosterlee, C. W. (2009). Pricing early-exercise and discrete barrier options by fourier-cosine series expansions. *Numerische Mathematik*, *114*(1), 27.

Fearnhead, P. (2002). Markov chain monte carlo, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics*, *11*(4), 848–862.

Fengler, M. R. (2006). *Semiparametric modeling of implied volatility.* Springer Science & Business Media.

Fengler, M. R. (2009). Arbitrage-free smoothing of the implied volatility surface. *Quantitative Finance*, *9*(4), 417–428.

Foresee, F. D., & Hagan, M. T. (1997). Gauss-newton approximation to bayesian learning. In *Neural networks, 1997., international conference on* (Vol. 3, pp. 1930–1935).

Fruhwirth-Schnatter, S. (1995). Bayesian model discrimination and bayes factors for linear gaussian state space models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 237–246.

Fulop, A., & Li, J. (2013). Efficient learning via simulation: A marginalized resample-move approach. *Journal of Econometrics*, *176*(2), 146–161.

Gatheral, J. (2011). *The volatility surface: a practitioner's guide* (Vol. 357). John Wiley & Sons.

Gençay, R., & Qi, M. (2001a). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *IEEE Transactions on Neural Networks*, *12*(4), 726–734.

Gençay, R., & Qi, M. (2001b). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *IEEE Transactions on Neural Networks*, *12*(4), 726–734.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Gilks, W. R., & Berzuini, C. (2001). Following a moving target?monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(1), 127–146.

Gong, B., Shi, Y., Sha, F., & Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *Computer vision and pattern recognition (cvpr), 2012 ieee conference on* (pp. 2066–2073).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Gourieroux, C., Monfort, A., & Renault, E. (1993). Indirect inference. *Journal of applied econometrics*, *8*(S1).

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on* (pp. 6645–6649).

Gujarati, D. N., & Porter, D. C. (1999). Essentials of econometrics.

Han, G.-S., & Lee, J. (2008). Prediction of pricing and hedging errors for equity linked warrants with gaussian process models. *Expert Systems with Applications*, *35*(1), 515–523.

Han, J., Zhang, X.-P., & Wang, F. (2016). Gaussian process regression stochastic volatility model for financial time series. *IEEE Journal of Selected Topics in Signal Processing*, *10*(6), 1015–1028.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Overview of supervised learning. In *The elements of statistical learning* (pp. 9–41). Springer.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, *6*(2), 327–343.

Heston, S. L., & Nandi, S. (2000). A closed-form garch option valuation model. *The Review of Financial Studies*, *13*(3), 585–625.

Hutchinson, J. M., Lo, A. W., & Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, *49*(3), 851–889.

Katsiampa, P. (2017). Volatility estimation for bitcoin: A comparison of garch models. *Economics Letters*, *158*, 3–6.

Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied soft computing*, *13*(2), 947–958.

Kim, S. (2009). The performance of traders' rules in options market. *Journal of Futures Markets*, *29*(11), 999–1020.

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (pp. 3581–3589).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kingman, J. (1970). Information theory and statistics. by kullback solomon. pp. 399. 28s. 6d. 1968.(dover.). *The Mathematical Gazette*, *54*(387), 90–90.

Kou, S. G. (2002). A jump-diffusion model for option pricing. *Management science*, *48*(8), 1086–1101.

Kou, S. G., & Wang, H. (2004). Option pricing under a double exponential jump diffusion model. *Management science*, *50*(9), 1178–1192.

Kristoufek, L. (2013). Bitcoin meets google trends and wikipedia: Quantifying the relationship between phenomena of the internet era. *Scientific reports*, *3*, 3415.

Kristoufek, L. (2015). What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. *PloS one*, *10*(4), e0123923.

Kwon, Y., & Lee, Y. (2011). A second-order tridiagonal method for american options under jump-diffusion models. *SIAM Journal on Scientific Computing*, *33*(4), 1860–1872.

Lajbcygier, P. R., & Connor, J. T. (1997). Improved option pricing using artificial neural networks and bootstrap methods. *International journal of neural systems*, *8*(04), 457–471.

Lattin, J. M., Carroll, J. D., & Green, P. E. (2003). *Analyzing multivariate data*. Thomson Brooks/Cole Pacific Grove, CA.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing american options by simulation: a simple least-squares approach. *The review of financial studies*, *14*(1), 113–147.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., & Winther, O. (2016). Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*.

MacKay, D. J. (1992). A practical bayesian framework for backpropagation networks. *Neural computation*, *4*(3), 448–472.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms.* Cambridge university press.

Madan, D. B., Carr, P. P., & Chang, E. C. (1998). The variance gamma process and option pricing. *Review of Finance*, *2*(1), 79–105.

Madan, D. B., & Seneta, E. (1990). The variance gamma (vg) model for share market returns. *Journal of business*, 511–524.

Madan, I., Saluja, S., & Zhao, A. (2015). *Automated bitcoin trading via machine learning algorithms.* nd.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Malliaris, M., & Salchenberger, L. (1996). Using neural networks to forecast the s&p 100 implied volatility. *Neurocomputing*, *10*(2), 183–195.

McNally, S. (2016). *Predicting the price of bitcoin using machine learning* (Unpublished doctoral dissertation). Dublin, National College of Ireland.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics*, *3*(1-2), 125–144.

Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*.

Minsky, M., & Papert, S. (1969). Perceptrons.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective.* MIT press.

Nadarajah, S., & Chu, J. (2017). On the inefficiency of bitcoin. *Economics Letters*, *150*, 6–9.

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.*

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies: A comprehensive introduction.* Princeton University Press.

Nualart, D., Schoutens, W., et al. (2001). Backward stochastic differential equations and feynman-kac formula for lévy processes, with applications in finance. *Bernoulli*, *7*(5), 761–776.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Papaspiliopoulos, O., Roberts, G. O., & Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, 59–73.

Park, H., & Lee, J. (2012). Forecasting nonnegative option price distributions using bayesian kernel methods. *Expert Systems with Applications*, *39*(18), 13243–13252.

Patricia, N., & Caputo, B. (2014). Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1442–1449).

Poyiadjis, G., Doucet, A., & Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, *98*(1), 65–80.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Ranganath, R., Tran, D., & Blei, D. (2016). Hierarchical variational models. In *International conference on machine learning* (pp. 324–333).

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian processes for machine learning* (Vol. 1). MIT press Cambridge.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Schmitz, A., Wang, Z., & Kimn, J.-H. (2014). A jump diffusion model for agricultural commodities with bayesian analysis. *Journal of Futures Markets*, *34*(3), 235–260.

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on signal Processing*, *50*(2), 281–289.

Tsitsiklis, J. N., & Van Roy, B. (2001). Regression methods for pricing complex american-style options. *IEEE Transactions on Neural Networks*, *12*(4), 694–703.

Urquhart, A. (2016). The inefficiency of bitcoin. *Economics Letters*, *148*, 80–82.

van Wijk, D. (2013). What can be expected from the bitcoin. *Erasmus Universiteit Rotterdam*.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Wang, P. (2011a). Pricing currency options with support vector regression and stochastic volatility model with jumps. *Expert Systems with Applications*, *38*(1), 1–7.

Wang, P. (2011b). Pricing currency options with support vector regression and stochastic volatility model with jumps. *Expert Systems with Applications*, *38*(1), 1–7.

Xiong, T., Bao, Y., & Hu, Z. (2014). Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems*, *55*, 87–100.

Xu, X., & Taylor, S. J. (1995). Conditional volatility and the informational efficiency of the phlx currency options market. *Journal of Banking & Finance*, *19*(5), 803–821.

Yang, S.-H., & Lee, J. (2011). Predicting a distribution of implied volatilities for option pricing. *Expert Systems with Applications*, *38*(3), 1702–1708.

Yao, J., & Tan, C. L. (2000a). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, *34*(1), 79–98.

Yao, J., & Tan, C. L. (2000b). Time dependent directional profit model for financial time series forecasting. In *Neural networks, 2000. ijcnn 2000, proceedings of the ieee-inns-enns international joint conference on* (Vol. 5, pp. 291–296).

Zhang, J. E., & Shu, J. (2003). Pricing s&p 500 index options with heston's model. In *Computational intelligence for financial engineering, 2003. proceedings. 2003 ieee international conference on* (pp. 85–92).

Table 5.1 Estimation performance. This table reports estimation errors for S&P 100 OEX put options of each category with respect to moneyness and time to maturity.

Panel A: Estimation Errors - Econometric jump models

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
Kou	<0.94	0.0278	-0.0105	1.894	3.0442	0.0444	-0.0276	2.6359	4.9841	0.1073	-0.0854	8.6396	12.0786
	(0.94, 0.97)	0.0306	-0.0033	0.853	1.1566	0.0399	-0.0267	1.2273	1.6835	0.0695	-0.022	2.4883	3.2805
	(0.97, 1.00)	0.0508	-0.0277	0.6726	1.0167	0.0573	-0.047	1.0258	1.3302	0.0734	-0.0493	1.6939	2.0761
	(1.00, 1.03)	0.0716	-0.0373	0.3647	0.6769	0.0533	-0.0179	0.5772	0.9411	0.0569	-0.0167	0.9443	1.397
	(1.03, 1.06)	0.0773	-0.0365	0.1862	0.3485	0.0768	0.0382	0.4641	0.7156	0.0796	0.0667	0.8434	1.2916
	>1.06	0.0952	-0.0351	0.147	0.2063	0.1189	0.059	0.3041	0.4893	0.1697	0.1612	0.6395	0.8983
CGMY	<0.94	0.0212	0.021	1.2526	1.9162	0.0331	0.0331	1.7228	3.6141	0.0505	0.0505	3.4651	12.6494
	(0.94, 0.97)	0.0282	-0.0008	0.7732	0.9156	0.032	0.0103	1.0014	1.6707	0.058	0.0562	2.1194	6.0563
	(0.97, 1.00)	0.089	-0.0825	1.0414	1.3986	0.0607	-0.0319	1.0123	1.5247	0.0712	0.0688	1.5821	4.1647
	(1.00, 1.03)	0.1522	-0.1421	0.6789	0.7296	0.0705	-0.0236	0.6969	0.8128	0.1073	0.1056	1.6609	4.2877
	(1.03, 1.06)	0.0965	-0.0688	0.2181	0.1293	0.0885	0.0376	0.5033	0.4701	0.1587	0.1585	1.6387	3.6163
	>1.06	0.099	-0.0368	0.1379	0.0331	0.1627	0.092	0.3969	0.3072	0.2413	0.2237	1.019	1.8226

Panel B: Estimation Errors - Machine learning models [Neural Network]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.0524	0.0088	2.9289	5.874	0.0111	0.0042	0.5453	1.1777	0.0116	0.0063	0.7185	2.2297
	(0.94, 0.97)	0.0329	0.0184	0.9758	2.8856	0.0324	0.0206	0.9686	3.8077	0.0139	0.0047	0.4711	1.2298
	(0.97, 1.00)	0.0377	-0.0096	0.4473	1.2668	0.0249	0.0056	0.4149	1.3744	0.0222	0.0054	0.5061	1.4014
	(1.00, 1.03)	0.1016	-0.0597	0.3547	1.1302	0.0334	0.0014	0.3074	1.0018	0.0543	0.0175	0.772	2.9164
	(1.03, 1.06)	0.1458	-0.0465	0.2676	1.0002	0.0514	-0.0071	0.2272	0.6358	0.0597	-0.028	0.6503	3.3211
	>1.06	0.2619	0.0322	0.3702	1.268	0.197	0.0453	0.3232	1.1182	0.2424	0.111	0.4911	1.8633
7day training	<0.94	0.1011	-0.0723	0.5521	0.7669	0.0573	-0.0071	0.4996	0.7176	0.1399	0.0758	0.5308	0.7094
	(0.94, 0.97)	0.0989	-0.0027	0.5526	0.8262	0.0965	-0.0206	0.5889	1.3294	0.0787	-0.0059	0.3855	0.5576
	(0.97, 1.00)	0.133	0.0216	0.5757	1.2986	0.1148	-0.0036	0.529	0.9818	0.0934	-0.0082	0.5183	0.8906
	(1.00, 1.03)	0.1116	0.0019	0.5405	1.1314	0.1273	0.0161	0.6251	1.1979	0.1013	-0.0042	0.6808	2.5971
	(1.03, 1.06)	0.1073	-0.0085	0.5045	0.7443	0.1467	-0.0009	0.5979	1.1082	0.1089	-0.0288	0.4779	0.8674
	>1.06	0.116	0.0132	0.595	1.0235	0.1443	-0.0052	0.6001	1.8473	0.1138	-0.0013	0.4838	1.0455
30day training	<0.94	0.1785	-0.023	0.9197	1.1508	0.1755	-0.0202	0.8536	1.0475	0.1186	0.0362	0.814	0.9626
	(0.94, 0.97)	0.2024	-0.0016	0.89	1.1408	0.1913	-0.0637	0.8061	1.0341	0.192	-0.1238	0.8427	1.0582
	(0.97, 1.00)	0.1719	-0.0092	0.8363	1.3245	0.16	-0.0229	0.7996	1.0412	0.1579	-0.0189	0.8084	1.0359
	(1.00, 1.03)	0.1633	-0.015	0.7967	1.0979	0.1708	-0.0139	0.8601	1.1062	0.1819	-0.0119	0.803	1.0186
	(1.03, 1.06)	0.1757	0.0007	0.8168	1.1084	0.1825	-0.0292	0.8912	1.2978	0.1626	-0.0521	0.7454	0.9638
	<0.94	0.1957	-0.0438	0.8854	1.1521	0.1683	-0.0303	0.834	1.052	0.1468	-0.0363	0.6909	0.9282

Panel C: Estimation Errors - Machine learning models [Bayesian Neural Network]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.0019	0.0011	0.1013	0.4351	0.0003	0.0001	0.0158	0.0364	0.0002	0	0.0117	0.0343
	(0.94, 0.97)	0.002	-0.0002	0.0579	0.1264	0.0016	0.0004	0.0504	0.1446	0.0004	0.0002	0.0133	0.0288
	(0.97, 1.00)	0.0066	0.0008	0.0708	0.142	0.0023	-0.0002	0.0398	0.0782	0.0026	-0.0012	0.0604	0.4396
	(1.00, 1.03)	0.0203	0.0024	0.0599	0.1109	0.0034	0.0001	0.0325	0.0509	0.0039	0.001	0.0623	0.4728
	(1.03, 1.06)	0.0317	-0.0053	0.0523	0.0975	0.0061	-0.001	0.0282	0.0497	0.0147	0.0037	0.1577	1.2249
	>1.06	0.04	0.0187	0.0547	0.1385	0.0115	0.003	0.0219	0.0753	0.0668	0.0532	0.1133	0.971
7day training	<0.94	0.0745	-0.0107	0.3213	0.396	0.0406	-0.0013	0.2409	0.2845	0.0569	0.0026	0.3332	0.4484
	(0.94, 0.97)	0.0702	0.0048	0.3367	0.4551	0.0597	-0.0098	0.2809	0.401	0.0456	0.007	0.2038	0.2514
	(0.97, 1.00)	0.0657	0.0012	0.2697	0.3653	0.0578	0.0051	0.2531	0.3662	0.0523	0.0069	0.2623	0.3898
	(1.00, 1.03)	0.0601	-0.0015	0.2751	0.4377	0.0595	-0.0072	0.279	0.3871	0.0624	0.0158	0.2543	0.3613
	(1.03, 1.06)	0.0557	-0.0008	0.2608	0.3426	0.0659	-0.0107	0.28	0.377	0.0694	-0.0048	0.2641	0.3434
	>1.06	0.0597	0.0344	0.3049	0.3953	0.062	-0.0086	0.2852	0.3959	0.0602	0.0081	0.237	0.3228
30day training	<0.94	0.1569	-0.0175	0.8309	1.0715	0.1458	-0.0247	0.7224	0.8804	0.0959	0.043	0.6294	0.7434
	(0.94, 0.97)	0.172	-0.0145	0.7793	0.9541	0.1759	-0.0763	0.7727	0.9751	0.1779	-0.0981	0.8235	1.0254
	(0.97, 1.00)	0.1473	-0.0054	0.7138	0.9357	0.1326	-0.0034	0.7274	0.9561	0.1436	-0.0276	0.7187	0.8967
	(1.00, 1.03)	0.1478	-0.0311	0.7268	0.9832	0.1492	-0.0248	0.7882	1.0286	0.1539	-0.0249	0.679	0.8882
	(1.03, 1.06)	0.1578	-0.0207	0.7561	1.0228	0.1571	-0.0309	0.7642	0.9726	0.1518	-0.0713	0.6622	0.8596
	<0.94	0.173	-0.0664	0.8299	1.1344	0.1542	-0.0205	0.7717	0.9764	0.1337	-0.0397	0.6388	0.8663

Panel D: Estimation Errors - Machine learning models [Support Vector Regression]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.1649	-0.1242	10.8973	17.0629	0.0964	0.0656	4.9298	5.8843	0.0918	0.0918	6.8432	7.2452
	(0.94, 0.97)	0.1149	0.1058	3.1546	4.141	0.1026	0.1026	3.1828	3.4521	0.0877	0.0877	3.1083	3.3248
	(0.97, 1.00)	0.1565	0.1516	2.0488	2.6373	0.0827	0.0824	1.4863	1.6916	0.0812	0.0794	1.8351	2.0358
	(1.00, 1.03)	0.2296	-0.1849	0.5893	0.8635	0.0348	0.0088	0.3698	0.5072	0.0576	0.032	0.9363	1.1398
	(1.03, 1.06)	0.2029	-0.2022	0.3474	0.4733	0.0725	-0.072	0.3771	0.4134	0.0593	-0.0374	0.6333	0.7504
	>1.06	0.1082	-0.0946	0.1692	0.2709	0.0787	-0.0618	0.2187	0.2706	0.0841	-0.0585	0.3491	0.4302
7day training	<0.94	0.2207	-0.1874	14.0542	21.1056	0.0919	0.0595	4.6561	5.5471	0.0915	0.0915	6.8172	7.2148
	(0.94, 0.97)	0.0916	0.0715	2.5568	3.1345	0.1003	0.1003	3.1121	3.3319	0.0867	0.0867	3.0809	3.263
	(0.97, 1.00)	0.1326	0.1314	1.7541	2.1579	0.0781	0.0779	1.4064	1.5938	0.0784	0.0768	1.7727	1.9814
	(1.00, 1.03)	0.1392	-0.097	0.4008	0.5544	0.0342	0.0068	0.3599	0.4922	0.0557	0.0291	0.8979	1.0919
	(1.03, 1.06)	0.1424	-0.1413	0.2651	0.3217	0.0717	-0.0709	0.3718	0.4075	0.0589	-0.0383	0.6294	0.7461
	>1.06	0.0901	-0.0733	0.1443	0.1877	0.0772	-0.0594	0.2145	0.2662	0.0842	-0.0597	0.3483	0.4288
30day training	<0.94	0.4404	-0.4035	26.5434	36.4711	0.087	0.0283	4.6922	6.1422	0.1258	0.1141	8.8903	10.3214
	(0.94, 0.97)	0.0949	0.0316	2.8261	4.1785	0.1058	0.1051	3.2826	3.6063	0.1119	0.1119	3.9685	4.3869
	(0.97, 1.00)	0.1114	0.1035	1.5091	1.8782	0.0806	0.074	1.4842	1.7509	0.0822	0.0779	1.9033	2.2462
	(1.00, 1.03)	0.1513	-0.0818	0.5669	0.7584	0.0788	-0.0082	0.7726	1.0117	0.0626	0.009	0.9899	1.2435
	(1.03, 1.06)	0.176	-0.1415	0.3518	0.5117	0.1404	-0.0903	0.6947	0.9432	0.0854	-0.0495	0.868	1.0513
	<0.94	0.151	-0.0447	0.2271	0.3938	0.1829	-0.079	0.412	0.6209	0.1484	-0.0973	0.5473	0.824

Table 5.2 1-day prediction performance. This table reports 1-day prediction errors of each model applied to each category of the S&P 100 OEX put options with respect to moneyness and time to maturity.

Panel A: 1-day prediction errors - Econometric jump models

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
Kou	<0.94	0.0358	-0.0171	2.4317	4.0791	0.0652	-0.0381	4.1835	9.0233	0.1994	-0.1711	16.0606	20.9225
	(0.94, 0.97)	0.0307	-0.0021	0.8562	1.1861	0.047	-0.0173	1.4492	1.9371	0.1141	-0.0584	3.9663	5.4025
	(0.97, 1.00)	0.0643	-0.0291	0.7984	1.1631	0.0697	-0.0483	1.2255	1.6389	0.1101	-0.0766	2.5557	4.3264
	(1.00, 1.03)	0.1366	-0.0543	0.6011	0.8986	0.0851	-0.024	0.8835	1.2479	0.0834	-0.0172	1.3537	1.9898
	(1.03, 1.06)	0.1638	-0.0436	0.3756	0.5522	0.12	0.0311	0.694	0.9691	0.1275	0.0534	1.4017	2.482
	>1.06	0.1694	-0.0544	0.2762	0.3782	0.174	0.0486	0.4308	0.6378	0.2271	0.1763	0.7887	1.0833
CGMY	<0.94	0.0222	0.0219	1.2963	2.0175	0.0339	0.0333	1.7635	3.8429	0.0556	0.0556	3.7003	14.6171
	(0.94, 0.97)	0.0319	0.0025	0.8859	1.1422	0.0435	0.0243	1.3732	3.0085	0.0787	0.0742	2.7885	10.3495
	(0.97, 1.00)	0.0984	-0.0849	1.1236	1.8418	0.0779	-0.0229	1.2961	3.0949	0.1026	0.0941	2.2945	10.6143
	(1.00, 1.03)	0.2065	-0.1525	0.8431	1.1993	0.1048	-0.0221	1.026	2.1661	0.1292	0.1204	1.9784	7.6153
	(1.03, 1.06)	0.18	-0.0744	0.3804	0.2853	0.1337	0.0362	0.7348	1.0345	0.1737	0.1637	1.8402	5.435
	>1.06	0.1876	-0.0387	0.2895	0.1591	0.2033	0.0773	0.4967	0.4721	0.3003	0.2798	1.1739	2.3922

Panel B: 1-day prediction errors - Machine learning models [Neural Network]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.21	0.1799	15.9497	26.6149	0.2027	0.1348	11.3933	17.1807	0.2518	0.2193	20.8573	31.9152
	(0.94, 0.97)	0.1076	0.0669	3.1541	5.4144	0.1255	0.0826	3.7916	5.9402	0.1814	0.0408	6.4249	9.4529
	(0.97, 1.00)	0.205	-0.0476	2.2153	4.3678	0.1302	0.0395	2.2922	4.0149	0.1995	0.0069	4.5879	9.8551
	(1.00, 1.03)	0.5855	-0.3886	1.7916	4.4946	0.1496	-0.001	1.4674	2.5154	0.2013	-0.0761	3.1786	6.6437
	(1.03, 1.06)	0.5454	-0.3173	1.04	3.9772	0.2335	-0.0577	1.1524	2.0578	0.3503	-0.0821	3.5839	7.6053
	>1.06	0.3725	-0.0898	0.5748	0.8901	0.6726	0.1431	1.2492	3.0115	1.9571	0.4261	4.786	10.0808
7day training	<0.94	0.0636	0.0353	4.9396	11.2759	0.0453	0.0132	3.3233	8.1083	0.1135	0.025	9.8222	16.2667
	(0.94, 0.97)	0.0377	0.0225	1.0953	1.8025	0.042	-0.021	1.2323	2.3237	0.0803	0.0219	2.732	7.3025
	(0.97, 1.00)	0.0682	-0.0059	0.7855	1.2673	0.0437	-0.0034	0.7493	1.0679	0.0829	0.0031	1.9716	6.8917
	(1.00, 1.03)	0.2094	-0.0475	0.7258	1.0436	0.0747	-0.0084	0.7491	1.0378	0.1044	0.0065	1.7213	5.7059
	(1.03, 1.06)	0.3263	0.1429	0.631	0.9339	0.1376	-0.0302	0.694	1.0723	0.1294	-0.0217	1.3576	3.3135
	>1.06	0.4072	0.171	0.5978	0.8681	0.2821	0.0011	0.5913	0.9449	0.558	-0.2874	1.3777	4.7696
30day training	<0.94	0.0489	-0.0003	3.7593	7.9338	0.0406	-0.0183	2.3686	3.9266	0.0847	0.0226	7.0436	11.3269
	(0.94, 0.97)	0.0312	0.0207	0.8899	1.2941	0.026	0.001	0.8338	1.1034	0.0368	0.0059	1.2629	1.6366
	(0.97, 1.00)	0.0677	-0.0181	0.7707	0.9713	0.0549	0.0006	0.9336	1.1417	0.0508	0.0025	1.1654	1.5102
	(1.00, 1.03)	0.2226	-0.068	0.879	1.1058	0.1098	-0.0209	1.0803	1.3301	0.084	-0.0177	1.3168	1.6834
	(1.03, 1.06)	0.3474	0.1238	0.7244	0.9214	0.1841	-0.0604	0.9733	1.2085	0.1209	-0.024	1.279	1.6121
	>1.06	0.4479	0.168	0.6873	0.8441	0.2808	-0.0836	0.6534	0.8699	0.2559	-0.099	0.831	1.1513

Panel C: 1-day prediction errors - Machine learning models [Bayesian Neural Network]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.119	0.0861	9.3025	18.6467	0.1175	0.1029	7.5052	13.7716	0.1724	0.1548	15.4529	23.8945
	(0.94, 0.97)	0.0381	0.0167	1.1402	2.6048	0.0465	0.0256	1.4516	2.5929	0.0637	0.0284	2.2602	3.6052
	(0.97, 1.00)	0.0946	-0.0266	1.0422	2.6307	0.0471	0.0087	0.8327	1.3957	0.0821	0.0074	1.9153	4.1176
	(1.00, 1.03)	0.3393	-0.2401	1.0815	3.0338	0.0608	-0.0003	0.6054	0.9028	0.1193	-0.0177	1.9891	7.3598
	(1.03, 1.06)	0.4542	-0.2848	0.8576	3.334	0.114	-0.0229	0.5795	1.0576	0.143	-0.0157	1.4997	3.0521
	>1.06	0.3512	-0.1788	0.5661	1.413	0.2244	-0.0644	0.4627	0.9171	0.8146	0.3513	1.9119	4.1309
7day training	<0.94	0.0685	0.0185	5.1722	10.7068	0.0446	0.0219	3.3056	8.3196	0.0472	0.0332	3.769	6.1329
	(0.94, 0.97)	0.0258	0.0098	0.7271	1.0496	0.0279	-0.005	0.852	1.0949	0.0328	0.0005	1.1322	1.3741
	(0.97, 1.00)	0.0479	0.0005	0.5711	0.8138	0.0399	0.0021	0.6911	1.0839	0.0459	0.0038	1.0701	2.5309
	(1.00, 1.03)	0.1579	0.008	0.6101	0.8411	0.0648	0.0012	0.6373	0.9479	0.0603	-0.0084	0.9812	2.0312
	(1.03, 1.06)	0.264	0.0781	0.5405	0.7539	0.1161	0.0014	0.6027	0.9005	0.0846	-0.0008	0.9017	1.3068
	>1.06	0.3512	-0.0052	0.5305	0.7171	0.2311	-0.004	0.5001	0.7657	0.2717	-0.0491	0.7606	2.0479
30day training	<0.94	0.0437	0.0003	3.2653	6.7632	0.0346	-0.0155	1.9639	2.9089	0.037	0.0187	3.3103	6.4997
	(0.94, 0.97)	0.0229	0.0103	0.6539	0.8229	0.0221	0.0025	0.6996	0.8975	0.0242	0.0057	0.834	1.0174
	(0.97, 1.00)	0.0587	-0.0074	0.6912	0.8717	0.0525	-0.0023	0.8891	1.102	0.0466	0.0043	1.0613	1.3707
	(1.00, 1.03)	0.1875	-0.0178	0.7861	0.9815	0.1088	-0.022	1.0699	1.3194	0.0813	-0.0204	1.2771	1.6086
	(1.03, 1.06)	0.3063	0.0153	0.6732	0.8201	0.1779	-0.0463	0.9573	1.2104	0.1223	-0.032	1.2843	1.5836
	>1.06	0.3208	-0.0501	0.5114	0.6449	0.2898	-0.0557	0.6742	0.8739	0.2663	-0.0963	0.8538	1.173

Panel D: 1-day prediction errors - Machine learning models [Support Vector Regression]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.5565	0.5565	34.1894	38.7797	0.6188	0.6188	33.0978	35.5899	0.7384	0.7384	56.0523	60.9195
	(0.94, 0.97)	0.5539	0.5539	15.9536	16.9894	0.5421	0.5421	16.9771	17.8847	0.6057	0.6057	21.3312	21.8791
	(0.97, 1.00)	0.3461	0.3227	4.9759	6.1339	0.4067	0.4067	7.3877	8.2056	0.5107	0.4961	11.6571	12.4385
	(1.00, 1.03)	0.6816	-0.5868	1.9405	2.5289	0.2267	0.1907	2.5217	3.2876	0.3833	0.3833	6.1477	6.966
	(1.03, 1.06)	1.3493	-1.3405	2.3928	2.8427	0.2778	-0.2074	1.2233	1.6028	0.2296	0.1861	2.5414	3.2734
	>1.06	1.7722	-1.7645	2.579	3.2177	0.9843	-0.9762	1.9043	2.3022	0.8849	-0.8354	2.1294	2.9783
7day training	<0.94	0.5482	0.4787	33.999	39.0737	0.6752	0.6752	36.3307	39.0568	0.7599	0.7599	57.7429	62.2533
	(0.94, 0.97)	0.557	0.557	16.1396	17.2888	0.5703	0.5703	17.9392	18.8421	0.6281	0.6281	22.214	22.7942
	(0.97, 1.00)	0.3564	0.3505	5.1198	6.2615	0.4215	0.4208	7.6604	8.4217	0.5258	0.5258	12.0006	12.6979
	(1.00, 1.03)	0.4279	-0.3229	1.4008	1.7632	0.2252	0.1834	2.5067	3.2323	0.393	0.393	6.3029	7.0434
	(1.03, 1.06)	1.1817	-1.179	2.2145	2.5433	0.3055	-0.2484	1.2893	1.5698	0.23	0.2002	2.5729	3.3045
	>1.06	1.8151	-1.8131	2.6541	3.0444	1.1235	-1.12	2.1663	2.4478	0.8945	-0.861	2.0332	2.5616
30day training	<0.94	0.4759	-0.3462	30.0725	47.8574	0.1909	0.1288	10.3105	14.6956	0.3041	0.2928	22.7543	27.893
	(0.94, 0.97)	0.168	0.1203	4.9338	7.1545	0.1745	0.172	5.5937	7.4795	0.2378	0.236	8.5043	10.1443
	(0.97, 1.00)	0.1637	0.1464	2.2887	3.1165	0.1336	0.1079	2.4801	3.4203	0.1686	0.1458	3.9705	5.18
	(1.00, 1.03)	0.2361	-0.1005	0.9959	1.3005	0.1485	-0.0033	1.5186	2.0073	0.1481	0.013	2.361	3.0999
	(1.03, 1.06)	0.3638	-0.2358	0.7747	1.0868	0.2502	-0.1169	1.3185	1.6978	0.2057	-0.0862	2.1006	2.7617
	>1.06	0.36	-0.0459	0.571	0.8221	0.3856	-0.1449	0.9046	1.2333	0.484	-0.37	1.495	2.1118

Panel E: 1-day prediction Errors - Machine learning models [Gaussian Process]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.8784	0.8784	53.6749	57.67	0.7947	0.7947	42.8487	45.756	0.7566	0.7566	57.0086	61.0158
	(0.94, 0.97)	0.7475	0.7475	21.4914	22.3438	0.6941	0.6941	21.6594	22.2121	0.5441	0.5342	19.3776	20.6764
	(0.97, 1.00)	0.5415	0.4615	7.4053	8.7898	0.4955	0.4811	8.9349	9.7016	0.4687	0.404	10.6798	11.945
	(1.00, 1.03)	1.0679	-0.7326	3.4442	5.0056	0.2879	0.0783	3.017	4.0643	0.327	0.1619	5.1054	6.6551
	(1.03, 1.06)	2.6842	-2.5774	4.7772	6.1232	0.7754	-0.753	3.5731	4.6825	0.3969	-0.251	3.9938	5.3412
	>1.06	3.9825	-3.9825	5.7914	6.6911	3.6029	-3.5921	6.997	7.8422	3.9821	-3.9244	9.3563	10.9986
7day training	<0.94	0.8836	0.8836	54.0669	58.0502	0.7989	0.7989	43.1346	46.1133	0.7312	0.7299	56.9973	63.3425
	(0.94, 0.97)	0.7878	0.7878	22.777	23.485	0.6883	0.686	21.6239	22.4436	0.5468	0.5468	19.5229	20.3444
	(0.97, 1.00)	0.567	0.5372	7.8396	8.9678	0.477	0.4688	8.6637	9.4659	0.4287	0.4147	9.8285	10.8459
	(1.00, 1.03)	0.6863	-0.412	2.3607	3.1383	0.2485	0.0624	2.5549	3.4198	0.269	0.0928	4.2492	5.6137
	(1.03, 1.06)	2.0313	-2.0152	3.6993	4.4336	0.8605	-0.8368	3.8484	4.6163	0.4375	-0.3631	4.2593	6.1436
	>1.06	3.4402	-3.4402	5.0787	5.5318	3.6549	-3.6323	7.07	7.655	4.3407	-4.3188	10.1976	11.7858
30day training	<0.94	0.1057	0.0578	8.6388	18.839	0.1845	0.1701	11.6497	19.0851	0.4834	0.4829	39.5147	46.9643
	(0.94, 0.97)	0.0621	0.0452	1.7633	4.3464	0.0903	0.0787	3.0004	4.3768	0.1974	0.1855	7.3123	10.1204
	(0.97, 1.00)	0.0893	0.0089	1.1288	1.9292	0.0842	0.003	1.4653	1.958	0.1492	0.0569	3.4445	5.3468
	(1.00, 1.03)	0.2319	-0.018	1.0544	1.389	0.1626	-0.0442	1.625	2.2533	0.1831	-0.0829	2.9148	4.9107
	(1.03, 1.06)	0.4945	-0.2495	1.0495	1.6355	0.2891	-0.0975	1.5732	2.5679	0.3087	-0.1866	3.2797	5.8616
	>1.06	0.6685	-0.0407	0.9922	1.4629	0.6913	-0.3951	1.4602	3.3745	2.3937	-2.1462	5.3416	9.7852

Table 5.3 7-day prediction performance. This table reports 7-day prediction errors of each model applied to each category of the S&P 100 OEX put options with respect to moneyness and time to maturity.

Panel A: 7-day prediction errors - Econometric jump models

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
Kou	<0.94	0.0287	-0.0062	1.8559	3.1454	0.057	-0.0274	3.6185	8.661	0.1767	-0.1344	14.2274	20.67
	(0.94, 0.97)	0.0322	0.0107	0.9041	1.1566	0.0496	-0.0137	1.5376	2.1501	0.1029	-0.0157	3.638	4.6215
	(0.97, 1.00)	0.0698	-0.0101	0.8614	1.2082	0.0712	-0.0315	1.2371	1.7326	0.0878	-0.0299	2.0377	3.0093
	(1.00, 1.03)	0.1769	-0.0379	0.7773	1.1126	0.1126	-0.0089	1.1321	1.5717	0.0974	0.0126	1.5817	2.2382
	(1.03, 1.06)	0.2246	-0.0198	0.5257	0.7713	0.1644	0.0296	0.9167	1.248	0.1508	0.0932	1.6572	2.2688
	>1.06	0.2639	-0.0124	0.4484	0.6188	0.2234	0.0432	0.5547	0.798	0.2512	0.1909	0.9224	1.2938
CGMY	<0.94	0.0223	0.0219	1.3003	2.0363	0.0334	0.0313	1.7414	3.7065	0.0334	0.0313	1.7414	3.7065
	(0.94, 0.97)	0.0306	0.0059	0.862	1.1502	0.0452	0.0165	1.4018	3.2398	0.0452	0.0165	1.4018	3.2398
	(0.97, 1.00)	0.0981	-0.0809	1.1234	1.9973	0.0909	-0.0304	1.5104	4.1254	0.0909	-0.0304	1.5104	4.1254
	(1.00, 1.03)	0.2326	-0.1538	0.9566	1.6203	0.1357	-0.031	1.3025	3.2168	0.1357	-0.031	1.3025	3.2168
	(1.03, 1.06)	0.249	-0.0699	0.5479	0.6007	0.1844	0.0204	0.9695	1.6953	0.1844	0.0204	0.9695	1.6953
	>1.06	0.3023	-0.0071	0.4824	0.4286	0.2549	0.0634	0.6135	0.7222	0.2549	0.0634	0.6135	0.7222

Panel B: 7-day prediction errors - Machine learning models [Neural Network]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.2791	0.2491	18.53	27.4221	0.2377	0.1526	13.4347	19.9674	0.3014	0.2525	23.9033	34.4332
	(0.94, 0.97)	0.2241	0.1464	6.4521	10.2144	0.2029	0.0705	6.1877	9.4778	0.2674	0.0575	9.5707	13.4074
	(0.97, 1.00)	0.312	-0.0319	3.7536	7.015	0.2535	-0.0297	4.3878	7.7949	0.2824	-0.0003	6.4088	11.2346
	(1.00, 1.03)	0.7816	-0.4784	2.8227	6.6589	0.3798	-0.1523	3.8409	7.8255	0.4038	-0.0853	6.3445	12.5211
	(1.03, 1.06)	1.1357	-0.6566	2.2181	6.3301	0.5747	-0.2686	3.0385	6.4017	0.5794	-0.1874	5.9628	12.0896
	>1.06	1.3193	-0.3566	2.1724	5.8656	1.4407	0.1847	2.9306	6.6172	2.5406	-0.184	6.3557	12.7917
7day training	<0.94	0.0985	0.0595	7.3141	13.9569	0.0467	-0.0004	2.998	5.8762	0.1084	0.066	9.4003	15.588
	(0.94, 0.97)	0.0529	0.0313	1.5398	2.5979	0.0385	-0.013	1.1882	1.824	0.0611	0.0081	2.1841	5.2758
	(0.97, 1.00)	0.0746	0.0022	0.9045	1.3628	0.0556	-0.0046	0.953	1.3641	0.0868	0.005	2.0632	6.602
	(1.00, 1.03)	0.2343	-0.0431	0.8808	1.2278	0.1048	-0.0178	1.0299	1.456	0.1294	-0.002	2.1495	6.3564
	(1.03, 1.06)	0.4365	0.1584	0.8617	1.2815	0.1824	-0.0392	0.9222	1.3114	0.1804	-0.0356	1.9862	5.3493
	>1.06	0.5864	0.1911	0.8564	1.3277	0.3905	-0.0437	0.8075	1.288	0.6787	-0.1175	1.701	4.7748
30day training	<0.94	0.0516	0.0005	4.0193	8.3099	0.0418	-0.022	2.3348	3.7079	0.0569	0.0254	4.8727	7.8367
	(0.94, 0.97)	0.0344	0.0236	0.9918	1.7367	0.0271	-0.0018	0.8571	1.1201	0.0367	0.0008	1.2818	1.6129
	(0.97, 1.00)	0.0726	-0.0184	0.8311	1.043	0.0588	-0.0002	1.0013	1.2201	0.0563	0.0063	1.2883	1.6563
	(1.00, 1.03)	0.2332	-0.0636	0.936	1.1762	0.1203	-0.0261	1.1816	1.4411	0.0928	-0.0183	1.4595	1.8435
	(1.03, 1.06)	0.3805	0.1218	0.794	1.0048	0.2039	-0.0688	1.0713	1.3186	0.1392	-0.0269	1.4836	1.8875
	>1.06	0.4607	0.157	0.7243	0.9024	0.3119	-0.0945	0.7168	0.9538	0.3001	-0.1233	0.9683	1.3718

Panel C: 7-day prediction errors - Machine learning models [Bayesian Neural Network]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.1629	0.1395	11.7581	19.828	0.1268	0.0661	7.8953	14.5095	0.2185	0.1758	18.3464	26.4381
	(0.94, 0.97)	0.0702	0.0454	2.0596	4.7313	0.0712	0.0067	2.2268	5.4176	0.1107	0.0114	3.9253	6.498
	(0.97, 1.00)	0.1298	-0.0253	1.4685	3.5643	0.0823	-0.0131	1.4429	3.2404	0.13	0.0148	2.9646	6.4824
	(1.00, 1.03)	0.466	-0.2992	1.6084	4.8412	0.1275	-0.0352	1.2938	2.8012	0.1708	-0.023	2.7013	7.2596
	(1.03, 1.06)	0.6886	-0.439	1.3982	4.6379	0.2145	-0.0828	1.1096	2.2845	0.227	-0.0429	2.3754	5.3437
	>1.06	0.964	-0.6747	1.4564	3.9757	0.5153	-0.0181	1.0022	2.4801	0.9053	0.2918	2.2245	4.7822
7day training	<0.94	0.0776	0.0353	5.6751	10.6748	0.0494	0.008	3.1227	6.0027	0.0836	0.0612	7.3751	12.7705
	(0.94, 0.97)	0.0377	0.0152	1.0781	1.7491	0.0506	-0.0042	1.5675	2.8633	0.0386	0.0094	1.3669	1.9043
	(0.97, 1.00)	0.0693	0.0095	0.847	1.2963	0.0717	-0.0039	1.2361	2.362	0.0635	-0.0004	1.4628	2.8867
	(1.00, 1.03)	0.2045	0.0187	0.8282	1.1731	0.1224	-0.0082	1.1939	2.2807	0.0999	-0.0184	1.6321	3.113
	(1.03, 1.06)	0.3748	0.1049	0.7674	1.1002	0.2306	-0.013	1.1608	2.3637	0.1248	-0.0172	1.3285	2.3527
	>1.06	0.5226	-0.0662	0.7784	1.1354	0.4947	-0.0146	1.0279	2.1984	0.3924	-0.0502	1.0892	2.2874
30day training	<0.94	0.0481	0.0036	3.701	7.3961	0.0379	-0.0148	2.2012	3.5899	0.0441	0.0218	3.8169	6.2202
	(0.94, 0.97)	0.0242	0.0114	0.6889	0.855	0.0266	0.0011	0.846	1.129	0.0315	0.0068	1.1011	1.3931
	(0.97, 1.00)	0.0636	-0.0066	0.7532	0.9508	0.0588	-0.0029	0.9972	1.2343	0.0524	0.0031	1.1912	1.5347
	(1.00, 1.03)	0.2046	-0.0193	0.8652	1.0777	0.1211	-0.0238	1.1945	1.459	0.0918	-0.0254	1.4361	1.8027
	(1.03, 1.06)	0.333	0.0117	0.7395	0.9064	0.2	-0.0514	1.0738	1.3384	0.1353	-0.0351	1.422	1.757
	>1.06	0.3751	-0.035	0.5999	0.7448	0.3254	-0.0566	0.7581	0.9765	0.2755	-0.123	0.9165	1.2594

Panel D: 7-day prediction errors - Machine learning models [Support Vector Regression]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.8544	0.8544	52.2725	56.4484	0.8201	0.8201	44.0138	46.8723	0.8802	0.8802	66.5222	70.9627
	(0.94, 0.97)	0.7585	0.7585	21.8429	22.5127	0.7534	0.7534	23.5437	24.1604	0.7599	0.7599	26.9243	27.4389
	(0.97, 1.00)	0.4975	0.4878	7.1518	8.3304	0.6146	0.6146	11.0744	11.723	0.7004	0.7004	15.9536	16.4406
	(1.00, 1.03)	0.7358	-0.565	2.2298	2.7424	0.3885	0.3576	4.3189	5.0817	0.5758	0.5758	9.1799	9.6941
	(1.03, 1.06)	1.8975	-1.8857	3.3312	3.7421	0.3984	-0.2191	1.7653	2.2602	0.381	0.3576	4.2074	4.8223
	>1.06	2.8463	-2.8437	4.1562	4.5046	1.7646	-1.7492	3.1937	3.6549	1.26	-1.1639	2.7422	3.3472
7day training	<0.94	0.8687	0.8687	53.3646	57.7891	0.864	0.864	46.3079	48.9604	0.9013	0.9013	67.8397	72.0189
	(0.94, 0.97)	0.7659	0.7654	22.1484	22.8282	0.7869	0.7869	24.6202	25.0508	0.8078	0.8078	28.6037	28.9294
	(0.97, 1.00)	0.5063	0.5005	7.2785	8.436	0.637	0.637	11.4882	12.0385	0.7274	0.7274	16.5745	16.9528
	(1.00, 1.03)	0.6012	-0.4455	1.9054	2.3575	0.3856	0.3613	4.3419	5.067	0.5925	0.5925	9.449	9.8507
	(1.03, 1.06)	1.8454	-1.8311	3.3576	3.6726	0.4249	-0.2706	1.7845	2.1415	0.3819	0.3757	4.2737	4.7999
	>1.06	3.1249	-3.1228	4.5067	4.6996	2.0226	-2.0145	3.6298	3.9287	1.3614	-1.2898	2.8371	3.3041
30day training	<0.94	0.7548	-0.647	54.0967	114.3841	0.1491	0.0846	8.4674	12.5218	0.3312	0.3195	24.8701	30.9839
	(0.94, 0.97)	0.1572	0.0996	4.615	6.687	0.1547	0.1536	4.8967	5.9983	0.2534	0.2514	9.0618	10.9615
	(0.97, 1.00)	0.1622	0.1443	2.2649	3.0155	0.1228	0.0984	2.2664	2.9049	0.1836	0.1626	4.335	5.593
	(1.00, 1.03)	0.2551	-0.1104	1.069	1.3913	0.1517	-0.0154	1.5251	1.9674	0.1552	0.0147	2.4869	3.4103
	(1.03, 1.06)	0.4094	-0.2476	0.8714	1.1903	0.2653	-0.1312	1.3908	1.7829	0.2152	-0.0942	2.2148	3.0275
	>1.06	0.4115	-0.026	0.6526	0.9024	0.4224	-0.1557	0.9897	1.3377	0.5658	-0.4405	1.6813	2.4594

Panel E: 7-day prediction errors - Machine learning models [Gaussian Process]

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
1day training	<0.94	0.8874	0.8874	54.2472	58.2472	0.7957	0.7957	42.9174	45.858	0.7691	0.7691	58.8784	63.7954
	(0.94, 0.97)	0.7656	0.7656	22.0789	22.915	0.693	0.693	21.6513	22.3169	0.5818	0.5773	20.8311	22.2846
	(0.97, 1.00)	0.5601	0.493	7.6661	8.9153	0.4837	0.4685	8.759	9.5931	0.4796	0.4323	10.9744	12.297
	(1.00, 1.03)	0.9409	-0.6081	3.0901	4.3652	0.2899	0.0556	2.9727	3.9601	0.3503	0.1464	5.5026	7.4174
	(1.03, 1.06)	2.3452	-2.2696	4.1813	5.4247	0.8483	-0.8173	3.83	4.9194	0.4616	-0.2763	4.628	6.7046
	>1.06	3.5762	-3.5693	5.2198	6.0382	3.73	-3.7128	7.2093	8.0567	3.8674	-3.8305	9.2484	11.1954
7day training	<0.94	0.891	0.891	54.3991	58.2823	0.8068	0.8068	43.3304	45.9388	0.7768	0.7766	59.7762	65.0325
	(0.94, 0.97)	0.7991	0.7991	23.0887	23.7415	0.7005	0.7	21.9458	22.5654	0.5908	0.5908	21.0267	21.8813
	(0.97, 1.00)	0.5727	0.5487	7.9261	9.0479	0.4749	0.4624	8.6408	9.4619	0.4445	0.4152	10.2396	11.3435
	(1.00, 1.03)	0.693	-0.4116	2.3952	3.1595	0.261	0.0445	2.6374	3.4888	0.2726	0.1107	4.2948	5.665
	(1.03, 1.06)	1.9797	-1.966	3.5603	4.3206	0.9059	-0.8774	3.9926	4.8088	0.4645	-0.37	4.5363	6.5584
	>1.06	3.2959	-3.2959	4.8318	5.2617	3.7432	-3.7223	7.1954	7.8091	4.1213	-4.1062	9.8402	11.211
30day training	<0.94	0.4453	-0.1643	2.2273	3.5043	0.3483	-0.0732	1.6413	2.711	0.5197	-0.3671	2.128	5.0487
	(0.94, 0.97)	0.4846	-0.2628	2.2094	6.2305	0.4582	-0.233	1.9736	4.4213	0.22	0.0991	1.8516	3.1959
	(0.97, 1.00)	0.4407	-0.2342	1.9762	4.9368	0.5087	-0.2086	1.8664	6.2717	0.3297	-0.0873	2.0017	5.3449
	(1.00, 1.03)	0.5737	-0.3262	2.3376	6.2033	0.5233	-0.3226	2.1791	7.3579	0.3669	-0.1867	1.8747	4.1665
	(1.03, 1.06)	0.5665	-0.2838	2.2765	5.5528	0.5408	-0.3629	2.0922	6.1125	0.4274	-0.2043	1.9381	4.4379
	>1.06	0.5644	-0.2328	2.1077	4.0584	0.506	-0.293	2.21	5.8981	0.4542	-0.1951	2.0004	6.5179

Table 5.4 1-day domain-adaptation performance. This table reports 1-day domain adaptation errors of each model with respect to moneyness and time to maturity. Each model is calibrated by European S&P 100 XEO put options and tested to predict 1-day ahead American S&P 100 OEX put options.

Panel A: 1-day domain-adaptation prediction errors - Econometric jump models

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
Kou	<0.94	0.0415	-0.0172	2.8852	4.6691	0.0637	-0.0391	3.2645	4.1738	0.1403	-0.0888	11.0735	13.8628
	(0.94, 0.97)	0.032	0.0054	0.8833	1.1078	0.086	-0.0213	2.659	3.1139	0.1184	0.0211	4.1735	4.6721
	(0.97, 1.00)	0.0857	-0.0541	1.004	1.278	0.108	-0.0529	1.8643	2.3923	0.1519	-0.0214	3.4206	4.4643
	(1.00, 1.03)	0.2097	-0.1487	0.8397	1.0838	0.1284	-0.0427	1.3353	1.7261	0.1434	-0.0072	2.2612	2.8422
	(1.03, 1.06)	0.2542	-0.1808	0.5196	0.6713	0.1648	-0.0442	0.8917	1.1263	0.138	0.0379	1.4914	1.9712
	>1.06	0.2217	-0.143	0.3381	0.4493	0.2417	-0.0718	0.557	0.7686	0.2516	0.1026	0.9242	1.3134
CGMY	<0.94	0.0208	0.02	1.2621	1.3629	0.0284	0.0223	1.5367	1.7804	0.0516	0.0516	3.5037	3.6344
	(0.94, 0.97)	0.0595	-0.0307	1.6161	2.169	0.0756	-0.019	2.298	3.2422	0.0835	0.0054	2.9263	3.7082
	(0.97, 1.00)	0.1983	-0.1838	2.3612	3.0244	0.1762	-0.1379	2.9674	3.8138	0.1405	-0.0199	3.1252	4.0165
	(1.00, 1.03)	0.4431	-0.4111	1.9239	2.5357	0.2748	-0.2272	2.7331	3.6206	0.1632	-0.0105	2.5342	3.4402
	(1.03, 1.06)	0.549	-0.4728	1.2329	1.7078	0.33	-0.2481	1.7721	2.4573	0.2132	-0.0177	2.2495	2.8941
	>1.06	0.5374	-0.431	0.8327	1.1671	0.4199	-0.2467	0.9909	1.4126	0.3308	0.0474	1.2085	1.6173

Panel B: 1-day domain-adaptation prediction errors - Machine learning models

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
Neural Networks	<0.94	0.4852	0.215	1.8605	3.4738	0.4374	-0.0441	3.3095	6.3626	0.3408	0.0118	2.3502	4.1223
	(0.94, 0.97)	0.5361	0.1069	2.4382	5.6013	0.2753	0.0417	2.0711	3.8329	0.8125	0.306	5.5506	15.1382
	(0.97, 1.00)	0.5325	-0.1577	2.4163	5.8453	0.6219	0.0484	2.6438	6.1134	0.476	-0.0739	2.6258	6.3891
	(1.00, 1.03)	0.4463	-0.0487	2.0212	4.2603	0.7305	0.0098	2.9575	7.3267	0.3738	-0.0843	1.8361	3.5174
	(1.03, 1.06)	0.5233	-0.2054	2.2933	5.9286	0.7586	-0.2062	3.0163	8.0315	0.6216	0.0773	2.2269	4.3203
	>1.06	0.4693	-0.2762	2.4655	7.7467	0.8679	-0.0438	3.1618	7.9471	0.6057	0.0055	2.3302	5.0105
Bayesian Neural Networks	<0.94	0.0821	0.0115	0.4675	0.6375	0.0855	-0.016	0.7811	1.7789	0.1813	0.0821	0.7438	0.954
	(0.94, 0.97)	0.1569	-0.0536	0.8	2.6112	0.0836	-0.0284	0.5285	0.7214	0.1658	0.0338	2.9696	12.9002
	(0.97, 1.00)	0.203	-0.0878	0.9447	3.4151	0.1691	-0.0066	0.8075	1.8435	0.1386	0.0096	0.7344	2.0524
	(1.00, 1.03)	0.1545	0.046	0.7051	1.6394	0.1385	-0.0331	0.738	1.6976	0.1409	-0.0539	0.7366	1.8697
	(1.03, 1.06)	0.1269	-0.0277	0.7361	2.5992	0.1439	-0.037	0.9124	3.4203	0.1393	-0.042	0.6742	1.437
	>1.06	0.1147	-0.047	1.0746	4.451	0.2386	-0.0673	1.2672	4.9817	0.1338	-0.0104	0.5728	1.0605
SVR	<0.94	0.5565	0.5565	34.1894	38.7797	0.6188	0.6188	33.0978	35.5899	0.7384	0.7384	56.0523	60.9195
	(0.94, 0.97)	0.5558	0.5558	16.0236	17.0531	0.5421	0.5421	16.9771	17.8847	0.6057	0.6057	21.3312	21.8791
	(0.97, 1.00)	0.3472	0.3235	4.9946	6.1536	0.4069	0.4069	7.385	8.2069	0.5103	0.4956	11.6404	12.4268
	(1.00, 1.03)	0.6835	-0.5876	1.9445	2.5347	0.227	0.1907	2.521	3.291	0.3833	0.3833	6.1477	6.966
	(1.03, 1.06)	1.3478	-1.3388	2.3956	2.849	0.2794	-0.209	1.2271	1.6081	0.2296	0.1861	2.5414	3.2734
	>1.06	1.772	-1.7643	2.5812	3.2216	0.9868	-0.9787	1.9064	2.3067	0.8882	-0.8382	2.137	2.9895
GP	<0.94	1.8925	-1.4586	7.8657	9.8611	1.3899	-0.9497	8.5331	11.6744	1.4715	-1.091	5.7829	6.9363
	(0.94, 0.97)	1.3877	-1.0262	7.056	10.9279	1.5283	-1.0371	7.9705	9.6931	2.14	-1.7744	11.3447	23.1693
	(0.97, 1.00)	1.5862	-1.1945	7.4774	11.0964	1.7382	-1.3656	7.1464	10.053	1.5182	-1.094	7.4924	10.9506
	(1.00, 1.03)	1.5261	-1.1401	6.9146	9.7897	1.6658	-1.2737	6.7597	9.3425	1.5088	-1.1135	6.7546	10.1175
	(1.03, 1.06)	1.7727	-1.4073	7.6277	11.5824	1.8989	-1.4686	7.4028	11.4598	1.6105	-1.2728	6.1856	8.9285
	>1.06	1.8044	-1.4188	8.7904	15.6597	1.7986	-1.2952	8.0795	12.4282	1.6033	-1.2915	6.5789	9.8638

Table 5.5 Estimation performance. This table reports the estimation error results for S&P 100 index American put options of each categories with respect to the moneyness,κ, and time to maturity,τ. Abbreviations as in Table 3.8.

Panel A: Estimation errors for pre-crisis period from 2003 to 2006.

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
GBNN	<0.94	0.0432	-0.0225	2.0501	4.3166	0.0311	-0.0062	1.4598	2.2041	0.0393	0.0366	2.0473	3.369
	(0.94, 0.97)	0.0371	-0.0142	0.9046	1.2579	0.0373	0.0035	0.9996	1.3385	0.0716	0.0613	2.2365	2.9398
	(0.97, 1.00)	0.0721	-0.0461	0.7097	0.8897	0.0669	-0.0168	0.9905	1.4377	0.0795	0.0278	1.5522	2.4128
	(1.00, 1.03)	0.2034	-0.142	0.7007	0.9109	0.1169	-0.0479	0.9153	1.3485	0.1083	0.0009	1.2678	2.0813
	(1.03, 1.06)	0.3121	-0.2091	0.6633	0.8327	0.1975	-0.1114	0.7963	1.199	0.1455	-0.023	1.0261	1.824
	>1.06	0.3303	-0.1085	0.5768	0.7326	0.2087	-0.1107	0.5471	0.8716	0.142	-0.0231	0.5462	1.0878
BNN	<0.94	0.0499	0.0492	2.232	3.3835	0.0348	0.0059	1.8748	4.2523	0.0315	-0.0141	1.5065	2.8001
	(0.94, 0.97)	0.0717	0.0627	1.719	2.2116	0.036	0.0054	0.9409	1.2563	0.0481	-0.0393	1.371	1.8739
	(0.97, 1.00)	0.1204	-0.0632	1.1282	1.4577	0.0417	-0.0212	0.5687	0.7902	0.067	-0.0484	1.1448	1.4985
	(1.00, 1.03)	0.661	-0.6605	1.8208	2.0136	0.0934	0.0046	0.5663	0.7104	0.1182	0.0836	1.0969	1.3806
	(1.03, 1.06)	0.9799	-0.9788	1.8266	2.0268	0.1798	-0.029	0.5697	0.7158	0.2622	0.2599	1.5551	1.7482
	>1.06	1.1163	-1.1159	1.8156	2.0016	0.2701	-0.1592	0.5417	0.6818	0.2816	0.1738	1.0367	1.3822
SVR	<0.94	0.1923	0.1585	8.9276	12.5162	0.1139	-0.0307	7.1747	16.2539	0.3045	-0.3032	17.0919	27.0151
	(0.94, 0.97)	0.3196	0.3196	7.767	8.256	0.1525	0.1421	3.9174	4.6701	0.1696	-0.1565	5.0245	5.872
	(0.97, 1.00)	0.2336	0.2008	2.8572	3.5579	0.1344	0.1274	1.8953	2.3052	0.128	-0.1088	2.3047	2.7411
	(1.00, 1.03)	0.5143	-0.4993	1.2091	1.4238	0.0692	-0.0042	0.4583	0.604	0.0904	-0.0773	1.0477	1.4033
	(1.03, 1.06)	0.8029	-0.8026	1.388	1.5241	0.1467	-0.13	0.3788	0.463	0.064	-0.0238	0.4208	0.5812
	>1.06	0.7109	-0.7096	1.1215	1.2772	0.1193	-0.0623	0.2531	0.3323	0.1397	0.125	0.344	0.4155
CGMY	<0.94	0.0285	0.0285	1.2374	1.478	0.0456	0.0455	1.8894	2.147	0.063	0.063	2.9753	3.0503
	(0.94, 0.97)	0.0477	0.0382	1.1671	1.4742	0.0775	0.0754	2.0721	2.4745	0.1141	0.1141	3.4234	3.9226
	(0.97, 1.00)	0.0793	-0.0484	0.7767	0.9166	0.0797	0.031	1.194	1.5424	0.1505	0.1476	2.9238	3.6847
	(1.00, 1.03)	0.2652	-0.1728	0.8881	1.1525	0.1024	0.0268	0.7978	1.119	0.1728	0.165	2.2503	2.8976
	(1.03, 1.06)	0.4781	-0.3969	0.8853	1.0763	0.1501	-0.0106	0.5465	0.7467	0.2214	0.2189	1.8122	2.2127
	>1.06	0.4767	-0.4303	0.6413	0.7225	0.1878	-0.0312	0.3697	0.5348	0.3224	0.3049	0.9952	1.262
AH-BS	<0.94	0.0296	0.0295	1.3348	1.4901	0.0473	0.0473	2.2483	2.4129	0.0792	0.0792	3.9122	4.1144
	(0.94, 0.97)	0.0411	0.0407	1.0155	1.1295	0.0581	0.0577	1.5385	1.6701	0.0976	0.0976	2.7945	2.926
	(0.97, 1.00)	0.0388	0.0251	0.446	0.569	0.0631	0.0606	0.8849	1.0175	0.1128	0.1126	1.9201	2.0874
	(1.00, 1.03)	0.081	0.0629	0.2508	0.3341	0.1027	0.0955	0.7115	0.8303	0.1443	0.1439	1.4675	1.6322
	(1.03, 1.06)	0.1756	0.1727	0.3461	0.4121	0.1638	0.1565	0.5689	0.6804	0.18	0.1794	1.1131	1.2672
	>1.06	0.2398	0.2395	0.4278	0.4949	0.2474	0.2451	0.542	0.6291	0.3012	0.301	0.8699	0.9845
AH-LV	<0.94	0.0272	0.027	1.228	1.4094	0.046	0.046	2.1826	2.3609	0.0802	0.0802	3.9711	4.1669
	(0.94, 0.97)	0.0414	0.0409	1.022	1.1457	0.0591	0.0588	1.5718	1.7152	0.0959	0.0959	2.7393	2.8563
	(0.97, 1.00)	0.0393	0.0254	0.4518	0.5776	0.0631	0.0606	0.8858	1.0216	0.1128	0.1126	1.9235	2.0903
	(1.00, 1.03)	0.082	0.0647	0.256	0.3436	0.1034	0.0953	0.7175	0.8371	0.1441	0.1435	1.4716	1.636
	(1.03, 1.06)	0.1782	0.1754	0.3555	0.4224	0.1654	0.1573	0.5781	0.6926	0.1784	0.1771	1.1054	1.2584
	>1.06	0.2502	0.2502	0.4447	0.5047	0.2514	0.2487	0.5531	0.6404	0.3013	0.3002	0.876	0.9944

Panel B: Estimation errors for crisis period from 2007 to 2009.

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
GBNN	<0.94	0.0186	0.0012	1.1153	2.8044	0.023	0.0128	1.4131	2.3509	0.0483	0.0257	3.5462	12.4067
	(0.94, 0.97)	0.0509	0.0183	1.568	2.4073	0.0633	0.0423	2.3464	3.4858	0.1104	0.1058	4.7986	6.2328
	(0.97, 1.00)	0.0948	-0.0145	1.4268	2.2677	0.0797	0.026	1.8804	3.2241	0.12	0.107	3.6562	5.695
	(1.00, 1.03)	0.2087	-0.0526	1.5074	2.4952	0.1204	0.0419	1.9186	3.5245	0.1461	0.1205	3.1853	5.539
	(1.03, 1.06)	0.2733	0.0538	1.7309	2.9942	0.1665	0.0449	1.8556	3.6922	0.1795	0.1401	2.8531	5.1525
	>1.06	0.2251	0.0128	0.9063	1.6997	0.1404	-0.0101	0.7843	1.8403	0.1746	0.0414	1.1708	2.6899
BNN	<0.94	0.0454	0.0322	2.9927	9.8176	0.0321	-0.0039	2.2144	3.7038	0.0466	-0.0322	3.5439	7.2872
	(0.94, 0.97)	0.0645	0.0193	1.8066	2.3923	0.0399	-0.0092	1.2971	1.7581	0.0639	-0.0403	2.4558	3.2808
	(0.97, 1.00)	0.131	-0.092	1.7554	2.2506	0.0417	-0.0063	0.8735	1.2486	0.0668	-0.0178	1.7399	2.2962
	(1.00, 1.03)	0.4901	-0.4893	2.5055	2.9001	0.0763	0.0215	0.9949	1.304	0.101	0.0552	1.8206	2.3695
	(1.03, 1.06)	0.8438	-0.8431	2.5482	2.8813	0.1496	0.052	1.2209	1.5573	0.2043	0.1966	2.4746	2.9874
	>1.06	0.9929	-0.9868	2.103	2.3793	0.2822	-0.0594	1.0112	1.3613	0.3342	0.2531	2.0816	2.6643
SVR	<0.94	0.1579	0.1057	11.8806	53.157	0.0982	-0.0124	8.5676	18.9792	0.1925	-0.1844	15.561	24.642
	(0.94, 0.97)	0.2364	0.2359	6.7342	7.6273	0.0971	0.0656	3.0429	3.9231	0.1756	-0.1667	6.7415	8.8147
	(0.97, 1.00)	0.1857	0.1647	2.7892	3.4401	0.1023	0.0825	1.9192	2.3131	0.1516	-0.1391	3.9713	5.2113
	(1.00, 1.03)	0.2768	-0.2341	1.119	1.4878	0.0652	0.0406	0.8508	1.0392	0.1267	-0.115	2.4327	3.3697
	(1.03, 1.06)	0.5675	-0.5639	1.4038	1.7501	0.0756	-0.0043	0.5584	0.7812	0.0784	-0.0538	1.1204	1.7399
	>1.06	0.5038	-0.5004	0.949	1.2724	0.1049	0.001	0.4172	0.5971	0.1064	0.0678	0.5525	0.8223
CGMY	<0.94	0.0267	0.0122	1.6877	2.0982	0.03	0.0229	2.0209	2.3645	0.0511	0.0507	3.7867	4.1299
	(0.94, 0.97)	0.0608	-0.0528	1.7347	2.2163	0.0517	-0.0285	1.772	2.1617	0.0716	0.0675	3.0169	3.5224
	(0.97, 1.00)	0.0988	-0.0813	1.3828	1.9259	0.0719	-0.0485	1.5158	2.0321	0.0683	0.0573	1.7477	2.3218
	(1.00, 1.03)	0.1517	-0.0037	1.0581	1.5175	0.0874	-0.027	1.1127	1.5676	0.1053	0.0848	1.7922	2.269
	(1.03, 1.06)	0.2364	0.0323	1.2582	1.7296	0.1128	-0.0038	0.9171	1.336	0.1318	0.1199	1.5712	2.1406
	>1.06	0.2315	0.0043	0.6664	0.9241	0.2169	0.0078	0.9432	1.2526	0.2054	0.1594	1.4086	1.9289
AH-BS	<0.94	0.026	0.0259	1.6982	2.0228	0.0348	0.0346	2.4286	2.8467	0.0604	0.0604	4.5801	4.872
	(0.94, 0.97)	0.0291	0.0266	0.8597	1.0984	0.042	0.0406	1.3767	1.6654	0.0832	0.083	3.1876	3.5791
	(0.97, 1.00)	0.0344	0.0186	0.5365	0.7712	0.0506	0.0467	1.0379	1.4504	0.0891	0.0877	2.2917	2.6988
	(1.00, 1.03)	0.0681	0.0442	0.4637	0.7233	0.0697	0.0653	0.9057	1.2041	0.1086	0.1074	1.9843	2.3941
	(1.03, 1.06)	0.1237	0.1131	0.506	0.7149	0.102	0.0917	0.8256	1.1759	0.1348	0.1338	1.7306	2.0906
	>1.06	0.2728	0.2672	0.7605	1.0505	0.2388	0.2336	0.8656	1.1873	0.2817	0.2812	1.4064	1.7447
AH-LV	<0.94	0.0263	0.0257	1.7286	2.0434	0.0352	0.0342	2.4613	2.8663	0.057	0.057	4.3306	4.6679
	(0.94, 0.97)	0.0292	0.0268	0.8608	1.1046	0.0419	0.04	1.3758	1.6859	0.0832	0.0831	3.1862	3.5651
	(0.97, 1.00)	0.0331	0.0189	0.5191	0.7473	0.0504	0.0463	1.0338	1.4651	0.0892	0.0881	2.2878	2.689
	(1.00, 1.03)	0.0654	0.0447	0.4415	0.6954	0.0697	0.0652	0.9088	1.2213	0.1089	0.108	2.0009	2.4198
	(1.03, 1.06)	0.122	0.115	0.4876	0.6915	0.102	0.0913	0.8236	1.1676	0.136	0.1354	1.7461	2.0996
	>1.06	0.2664	0.2623	0.7358	1.0304	0.2394	0.2349	0.8657	1.1825	0.2841	0.2837	1.4123	1.7355

Panel C: Estimation errors for post-crisis period from 2010 to 2012.

Model	moneyness	< 30 MAPE	< 30 MPE	< 30 MAE	< 30 RMSE	30 - 60 MAPE	30 - 60 MPE	30 - 60 MAE	30 - 60 RMSE	> 60 MAPE	> 60 MPE	> 60 MAE	> 60 RMSE
GBNN	<0.94	0.0142	0.0011	0.699	1.2622	0.0272	0.0209	1.3108	1.7498	0.0598	0.0592	4.0803	9.9194
	(0.94, 0.97)	0.0295	0.0137	0.8024	1.0614	0.0442	0.0325	1.3573	1.9345	0.11	0.1084	3.8576	4.486
	(0.97, 1.00)	0.059	-0.0096	0.712	0.9835	0.0668	0.0133	1.2178	1.7592	0.099	0.0836	2.3305	3.0747
	(1.00, 1.03)	0.1652	-0.009	0.7866	1.1327	0.1082	0.0232	1.1973	1.8269	0.1176	0.0848	1.9352	2.6228
	(1.03, 1.06)	0.2507	0.049	0.712	1.1029	0.1632	0.0254	1.0838	1.7795	0.1703	0.1371	2.0209	2.8224
	>1.06	0.2366	-0.0214	0.504	0.8033	0.1622	-0.0134	0.5444	1.0498	0.1575	0.0278	0.8515	1.8406
BNN	<0.94	0.05	0.0476	2.4742	3.6556	0.0303	-0.0068	1.615	2.1377	0.0365	-0.0307	2.0772	2.9669
	(0.94, 0.97)	0.0765	0.0578	2.1035	3.0063	0.0394	-0.0015	1.1665	1.6137	0.0551	-0.0457	1.9378	2.4294
	(0.97, 1.00)	0.1173	-0.0649	1.4014	1.8256	0.0448	-0.0182	0.8303	1.1134	0.0631	-0.0422	1.4769	1.86
	(1.00, 1.03)	0.5217	-0.5194	1.9733	2.2701	0.0722	0.0116	0.761	0.9788	0.0843	0.0268	1.4081	1.83
	(1.03, 1.06)	0.8016	-0.7999	1.8654	2.1437	0.1522	0.0661	0.9732	1.1804	0.193	0.1815	2.1563	2.5229
	>1.06	0.878	-0.8772	1.6083	1.8284	0.2629	-0.0513	0.7514	0.9918	0.3242	0.2329	1.7139	2.2036
SVR	<0.94	0.1813	0.1513	9.4817	11.8683	0.1135	-0.0022	7.0781	14.4051	0.3661	-0.3601	29.0676	48.9946
	(0.94, 0.97)	0.3045	0.3045	8.3792	9.3376	0.1341	0.0974	3.9424	4.9464	0.2033	-0.1818	7.3392	9.3845
	(0.97, 1.00)	0.2333	0.2186	3.1507	3.8622	0.1203	0.0871	2.116	2.6079	0.1916	-0.1796	4.5675	6.2556
	(1.00, 1.03)	0.337	-0.2704	1.0452	1.3193	0.0673	0.0334	0.7338	0.9487	0.1635	-0.1509	2.756	3.8344
	(1.03, 1.06)	0.6045	-0.6017	1.1756	1.3948	0.0681	-0.0351	0.367	0.4719	0.0997	-0.077	1.2008	1.8079
	>1.06	0.476	-0.4732	0.8253	1.0552	0.1091	-0.0053	0.3074	0.4111	0.1419	0.0993	0.5386	0.8213
CGMY	<0.94	0.0428	0.0428	1.89	1.9551	0.0661	0.0661	3.3556	3.5496	0.094	0.094	5.1599	5.5079
	(0.94, 0.97)	0.0545	0.0543	1.4427	1.6613	0.0862	0.0826	2.5853	3.0672	0.1593	0.1593	5.3671	6.1134
	(0.97, 1.00)	0.0854	0.0018	1.0734	1.3212	0.0845	0.0479	1.6071	2.1584	0.1806	0.1792	4.3995	5.1592
	(1.00, 1.03)	0.2321	-0.1012	1.1039	1.4503	0.1089	0.0493	1.2555	1.6824	0.2106	0.1973	3.5683	4.1054
	(1.03, 1.06)	0.3738	-0.2033	1.0776	1.3756	0.1455	0.0402	1.0105	1.3535	0.2385	0.2303	2.8168	3.2279
	>1.06	0.4176	-0.2405	0.7515	0.9458	0.2185	0.0327	0.6312	0.9519	0.3668	0.3544	1.6124	2.0559
AH-BS	<0.94	0.0282	0.0276	1.5168	1.6889	0.0397	0.039	2.2003	2.4595	0.0683	0.0683	4.3869	4.5748
	(0.94, 0.97)	0.0305	0.0292	0.8475	0.9832	0.0439	0.042	1.322	1.5369	0.0848	0.0848	2.957	3.3202
	(0.97, 1.00)	0.0325	0.0169	0.4394	0.57	0.0527	0.0478	0.982	1.2619	0.1019	0.1017	2.3986	2.7404
	(1.00, 1.03)	0.0732	0.0599	0.3658	0.5023	0.0803	0.0732	0.8988	1.1255	0.1257	0.1257	2.0665	2.3748
	(1.03, 1.06)	0.1584	0.1516	0.4485	0.5688	0.1198	0.1102	0.7953	0.979	0.1546	0.1543	1.7769	2.0226
	>1.06	0.2857	0.2804	0.6126	0.7632	0.2717	0.2679	0.7361	0.8718	0.3534	0.3529	1.3934	1.6065
AH-LV	<0.94	0.0273	0.027	1.4714	1.6247	0.041	0.0409	2.2802	2.526	0.0655	0.0655	4.1798	4.3584
	(0.94, 0.97)	0.0301	0.0291	0.8374	0.9583	0.0458	0.0454	1.3826	1.5789	0.0865	0.0865	3.0157	3.3559
	(0.97, 1.00)	0.0309	0.0179	0.4224	0.5468	0.0542	0.051	1.0068	1.2726	0.1013	0.1009	2.3819	2.6956
	(1.00, 1.03)	0.0729	0.0601	0.3599	0.4842	0.083	0.0773	0.9258	1.1468	0.1272	0.1267	2.0947	2.3939
	(1.03, 1.06)	0.1574	0.151	0.4415	0.5504	0.1228	0.1144	0.8138	0.9922	0.1559	0.1556	1.7903	2.0349
	>1.06	0.2808	0.2764	0.5961	0.7219	0.2752	0.272	0.7443	0.8723	0.3541	0.3534	1.3999	1.615

Table 5.6 Prediction performance. Table reports the prediction error results for S&P 100 index American put options of each categories with respect to the moneyness, κ, and time to maturity, τ. Abbreviations as in Table 3.8.

Panel A:Prediction errors for pre-crisis period from 2003 to 2006 for machine learning models

Model	moneyness	< 30 MAPE	< 30 MPE	< 30 MAE	< 30 RMSE	30 - 60 MAPE	30 - 60 MPE	30 - 60 MAE	30 - 60 RMSE	> 60 MAPE	> 60 MPE	> 60 MAE	> 60 RMSE
GBNN	<0.94	0.0516	0.0316	3.0706	6.2288	0.0633	0.0578	2.987	3.9503	0.0926	0.0926	4.5942	5.1795
	(0.94, 0.97)	0.0355	0.0141	0.8813	1.1486	0.0428	0.0263	1.1752	1.6039	0.0803	0.0732	2.4222	3.15
	(0.97, 1.00)	0.0787	-0.0537	0.7293	0.9083	0.0598	-0.0255	0.7953	1.0159	0.0817	0.0238	1.488	2.1302
	(1.00, 1.03)	0.2357	-0.1867	0.7008	0.8891	0.1101	-0.067	0.7104	0.8857	0.1091	-0.0156	1.101	1.4523
	(1.03, 1.06)	0.3334	-0.275	0.5682	0.6917	0.1869	-0.1298	0.5757	0.7201	0.1537	-0.0433	0.8838	1.1881
	>1.06	0.332	-0.314	0.4249	0.4952	0.2044	-0.1189	0.3805	0.5012	0.1935	-0.0394	0.5313	0.7754
BNN	<0.94	0.1121	0.1101	5.8627	10.3872	0.1128	0.0778	5.884	9.1686	0.1582	0.0696	8.4295	12.2255
	(0.94, 0.97)	0.1063	0.0922	2.6617	3.7587	0.0765	0.0294	2.0443	3.1091	0.119	-0.0655	3.4279	4.7161
	(0.97, 1.00)	0.1407	-0.0679	1.3409	1.7408	0.0609	-0.0201	0.8439	1.2144	0.0988	-0.0743	1.7102	2.5124
	(1.00, 1.03)	0.6972	-0.6961	1.8994	2.1532	0.1025	0.0021	0.6361	0.8304	0.1337	0.0825	1.269	1.6114
	(1.03, 1.06)	0.9999	-0.9995	1.8505	2.0987	0.1873	-0.0353	0.5858	0.7665	0.2787	0.2744	1.6653	1.8784
	>1.06	1.1201	-1.1201	1.821	2.0309	0.2841	-0.1719	0.5548	0.7017	0.3002	0.1799	1.0939	1.4491
SVR	<0.94	0.2222	0.1678	11.217	20.7542	0.1685	-0.0377	10.7219	22.5889	0.4235	-0.4209	23.8159	35.8442
	(0.94, 0.97)	0.3442	0.3439	8.4336	9.2167	0.1771	0.1572	4.6068	5.6636	0.2152	-0.2023	6.3586	7.8611
	(0.97, 1.00)	0.2732	0.2062	3.1635	3.9534	0.1642	0.1271	2.2791	2.8718	0.1564	-0.1233	2.7783	3.7322
	(1.00, 1.03)	0.6403	-0.5081	1.638	2.065	0.2199	-0.0041	1.3323	1.6711	0.1732	-0.0933	1.7533	2.3847
	(1.03, 1.06)	0.9645	-0.8467	1.7334	2.1494	0.4446	-0.1924	1.2536	1.5806	0.2573	-0.0259	1.386	1.7492
	>1.06	0.9431	-0.7408	1.5037	1.9424	0.57	-0.1856	1.119	1.4236	0.4458	0.0091	1.1439	1.4292
GP	<0.94	0.5036	0.4959	24.2586	31.0885	0.5798	0.5751	30.2097	37.9703	0.5563	0.5477	28.6108	34.6426
	(0.94, 0.97)	0.4439	0.432	11.0012	13.2956	0.4821	0.4762	12.8917	15.2172	0.5028	0.4985	14.5117	16.9289
	(0.97, 1.00)	0.292	0.0992	3.2863	4.5297	0.3032	0.2788	4.5019	5.8224	0.3722	0.3617	6.5392	8.0794
	(1.00, 1.03)	1.4322	-1.4046	3.544	4.5841	0.3487	-0.2275	1.9207	2.6271	0.2304	0.1096	2.5042	3.4925
	(1.03, 1.06)	2.7738	-2.7505	4.9971	6.2609	1.2802	-1.2557	3.2348	4.1626	0.4478	-0.3472	2.1344	2.8711
	>1.06	4.3286	-4.3041	6.9219	8.7219	2.6932	-2.6569	5.0124	6.4443	1.9893	-1.9336	4.339	5.8108

Panel A:Prediction errors for pre-crisis period from 2003 to 2006 for econometric models

Model	moneyness	< 30				30 - 60				> 60			
		MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
CGMY	<0.94	0.0295	0.0286	1.2587	1.5308	0.0432	0.0432	1.8787	2.2171	0.059	0.0589	2.8142	3.1713
	(0.94, 0.97)	0.0556	0.0035	1.3516	1.6933	0.0711	0.0166	1.8539	2.2833	0.1021	0.0538	2.9074	3.3877
	(0.97, 1.00)	0.2197	-0.2052	2.0202	2.5361	0.1699	-0.1271	2.1074	2.925	0.1671	-0.0026	2.7436	3.3352
	(1.00, 1.03)	0.6663	-0.6505	2.0545	2.4172	0.3428	-0.3091	2.073	2.6242	0.2143	-0.0744	2.0099	2.6887
	(1.03, 1.06)	1.0418	-1.0371	1.7676	1.9778	0.4974	-0.4734	1.4997	1.7675	0.2979	-0.1802	1.5602	2.0606
	>1.06	0.8453	-0.842	1.107	1.2455	0.5311	-0.4645	0.9702	1.1512	0.3054	-0.0705	0.8308	1.1004
LSM-BS	<0.94	0.1586	0.1586	6.8626	7.8223	0.341	0.3288	15.8048	16.8546	0.5089	0.5089	24.8947	25.8825
	(0.94, 0.97)	0.2274	0.2136	5.6172	6.5575	0.4624	0.447	12.2508	13.3381	0.6225	0.6225	17.8195	18.1955
	(0.97, 1.00)	0.2666	0.0399	2.768	3.6834	0.4779	0.4616	6.826	7.9092	0.6908	0.6815	11.8386	12.5091
	(1.00, 1.03)	0.8743	-0.7226	2.2901	3.4662	0.5002	0.4178	3.7728	5.256	0.7397	0.7027	7.6989	8.9338
	(1.03, 1.06)	0.915	-0.655	1.6627	2.4382	0.611	0.3987	2.3115	4.2106	0.8328	0.7283	5.2458	7.6261
	>1.06	0.7037	0.0518	1.286	1.6231	0.7734	0.5996	1.895	3.0366	0.9283	0.8427	3.0228	4.3476
LSM-GARCH	<0.94	0.0266	-0.0107	1.1232	1.35	0.0431	0.0351	1.9101	2.6404	0.0669	0.0654	3.1647	4.2773
	(0.94, 0.97)	0.071	-0.0383	1.6481	1.9522	0.082	0.0319	2.1776	3.2141	0.1293	0.1201	3.9258	5.5344
	(0.97, 1.00)	0.2548	-0.1969	2.3516	2.7137	0.1888	-0.0369	2.5908	3.4266	0.1808	0.1205	3.483	4.924
	(1.00, 1.03)	0.6252	-0.4546	1.8762	2.3304	0.3494	-0.0945	2.4151	3.2656	0.2656	0.1824	3.2518	4.6147
	(1.03, 1.06)	0.5252	0.31	1.2967	1.859	0.3986	0.1301	1.774	2.731	0.3909	0.3634	3.0678	4.2726
	>1.06	0.9136	0.9136	1.799	2.1057	0.7044	0.6871	1.8707	2.4755	0.6918	0.6883	2.4585	3.2397
LSM-Heston	<0.94	0.1932	-0.1932	8.371	8.7738	0.1476	-0.1441	6.7357	7.6602	0.1348	-0.1227	6.3908	7.45
	(0.94, 0.97)	0.4073	-0.4072	9.6033	10.1801	0.2986	-0.2821	7.6246	8.5522	0.2345	-0.1773	6.5193	7.6011
	(0.97, 1.00)	1.1113	-1.1078	9.9483	10.6636	0.5266	-0.4833	6.5166	7.4865	0.3187	-0.2007	5.0364	6.1554
	(1.00, 1.03)	3.8604	-3.8556	9.6732	10.3516	1.1485	-1.096	6.1589	7.0468	0.5462	-0.4137	4.6964	5.8291
	(1.03, 1.06)	3.909	-3.9015	6.4014	6.9228	2.1531	-2.1054	5.3202	6.0481	0.9074	-0.7735	4.1835	5.119
	>1.06	2.2193	-2.1998	3.1492	3.8159	1.6578	-1.4851	2.947	3.7	1.0191	-0.6778	2.5261	3.3393
B-W	<0.94	0.0384	0.038	1.5726	1.9217	0.0655	0.0653	2.8746	3.5316	0.0829	0.0829	3.8895	4.9433
	(0.94, 0.97)	0.0744	0.0731	1.7779	2.3367	0.1277	0.1275	3.3935	4.4967	0.1957	0.1957	5.7579	7.1289
	(0.97, 1.00)	0.1368	0.0295	1.4218	2.0225	0.2702	0.2701	3.965	4.8007	0.4162	0.4162	7.2981	8.0491
	(1.00, 1.03)	0.4678	-0.3175	1.3353	1.8207	0.4028	0.3958	3.2391	4.1027	0.5994	0.5994	6.3957	7.0419
	(1.03, 1.06)	0.3773	-0.0478	0.8874	1.4145	0.5175	0.5068	2.1898	2.9565	0.7338	0.7338	4.7672	5.4334
	>1.06	0.5196	0.4532	1.1348	1.5678	0.7313	0.7307	1.8795	2.4111	0.8874	0.8874	2.96	3.5925
AH-BS	<0.94	0.1719	0.1486	8.7434	19.4273	0.0913	0.0718	5.2225	15.1669	0.0718	0.0535	3.5752	4.2163
	(0.94, 0.97)	0.1037	0.0532	2.503	3.7535	0.0975	0.0523	2.5404	3.3184	0.1038	0.0727	2.9561	3.6672
	(0.97, 1.00)	0.2182	0.0106	2.1203	2.6849	0.1633	0.0542	2.1613	2.7294	0.1626	0.1116	2.6737	3.3358
	(1.00, 1.03)	0.605	-0.2514	1.8648	2.3602	0.3076	0.0369	1.9623	2.4348	0.2419	0.1301	2.2999	2.8466
	(1.03, 1.06)	0.9024	-0.5541	1.7506	2.2351	0.556	-0.1563	1.7886	2.2664	0.3621	0.0659	2.0386	2.5201
	>1.06	1.0038	-0.6807	1.7308	2.2068	0.7826	-0.4178	1.6149	2.0725	0.6147	-0.1801	1.6874	2.1443
AH-LV	<0.94	0.1578	0.1382	8.2324	18.7593	0.0863	0.0707	5.0624	14.928	0.1058	0.0999	5.2498	9.5583
	(0.94, 0.97)	0.1119	0.0565	2.69	4.1213	0.1016	0.0605	2.6558	3.4657	0.1228	0.1002	3.5329	5.578
	(0.97, 1.00)	0.2194	0.0217	2.145	2.7046	0.1707	0.0592	2.246	2.8179	0.1685	0.1173	2.7981	3.824
	(1.00, 1.03)	0.5552	-0.1811	1.7881	2.263	0.3158	0.0537	2.0141	2.5295	0.2549	0.1402	2.473	3.2486
	(1.03, 1.06)	0.814	-0.4042	1.6051	2.0702	0.5434	-0.1125	1.7531	2.2228	0.3739	0.093	2.1275	2.686
	>1.06	0.8919	-0.4601	1.5655	2.0238	0.7766	-0.375	1.6102	2.1024	0.6272	-0.1447	1.7869	2.2844

Panel B: Prediction errors for the financial crisis period from 2007 to 2009 for machine learning models

Model	moneyness	Days to Expiration < 30 MAPE	< 30 MPE	< 30 MAE	< 30 RMSE	30 - 60 MAPE	30 - 60 MPE	30 - 60 MAE	30 - 60 RMSE	> 60 MAPE	> 60 MPE	> 60 MAE	> 60 RMSE
GBNN	<0.94	0.0503	0.002	4.7952	14.0831	0.0537	0.0179	5.0655	13.6691	0.0639	0.0304	6.2403	12.1361
	(0.94, 0.97)	0.056	0.0205	1.7375	2.7216	0.0729	0.0507	2.7313	4.217	0.1189	0.1145	5.1831	6.7687
	(0.97, 1.00)	0.1038	-0.0133	1.583	2.5514	0.0876	0.0316	2.1068	3.7655	0.1267	0.1139	3.8848	6.0858
	(1.00, 1.03)	0.225	-0.0531	1.6538	2.7877	0.1325	0.0475	2.1509	4.0351	0.1524	0.1245	3.3478	5.8341
	(1.03, 1.06)	0.2844	0.0638	1.8441	3.2462	0.1812	0.0549	2.0621	4.1258	0.1871	0.1467	2.9772	5.3678
	>1.06	0.3276	0.0305	1.3431	2.1442	0.2164	-0.006	1.1717	2.3269	0.2305	0.0371	1.5594	3.1191
BNN	<0.94	0.0879	0.0669	7.1208	18.5315	0.0832	0.0065	6.944	16.4307	0.1408	-0.0368	13.0985	25.8951
	(0.94, 0.97)	0.0883	0.0375	2.5762	3.7873	0.0661	-0.0035	2.1652	3.3163	0.1228	-0.0617	4.715	6.358
	(0.97, 1.00)	0.1515	-0.0984	2.0253	2.6311	0.0621	-0.0042	1.2841	1.8253	0.1016	-0.0374	2.6137	3.57
	(1.00, 1.03)	0.5373	-0.5286	2.6714	3.1628	0.0959	0.0205	1.2565	1.7364	0.1182	0.0556	2.1755	2.9178
	(1.03, 1.06)	0.8735	-0.8661	2.6353	3.0326	0.1669	0.0482	1.3743	1.8439	0.2205	0.2054	2.7096	3.3007
	>1.06	1.013	-1.0018	2.2048	2.5223	0.3138	-0.0717	1.1226	1.5992	0.3569	0.2581	2.192	2.8514
SVR	<0.94	0.1961	0.111	15.6872	39.2062	0.1396	-0.038	13.6099	42.9998	0.3233	-0.3089	30.0887	68.1736
	(0.94, 0.97)	0.2596	0.2544	7.4681	8.9884	0.1275	0.0756	4.0589	5.2037	0.2336	-0.216	9.0442	13.3749
	(0.97, 1.00)	0.2522	0.167	3.6134	4.5428	0.1563	0.0862	3.007	3.7759	0.2125	-0.1741	5.4776	7.8047
	(1.00, 1.03)	0.4877	-0.2512	2.3852	2.9744	0.2069	0.0314	2.3152	2.9178	0.2063	-0.1371	3.6202	5.0251
	(1.03, 1.06)	0.8426	-0.6284	2.4687	3.096	0.3346	-0.0337	2.1748	2.757	0.2249	-0.0611	2.4997	3.2787
	>1.06	0.9608	-0.6789	2.0281	2.5976	0.6166	-0.1947	1.8907	2.4368	0.4941	-0.103	2.0595	2.6595
GP	<0.94	0.483	0.4641	37.87	58.0566	0.4892	0.4732	41.076	62.0236	0.5068	0.4903	47.0684	67.0817
	(0.94, 0.97)	0.3426	0.3071	10.1073	13.19	0.3898	0.3657	12.8051	15.7819	0.4241	0.4091	16.1671	19.7867
	(0.97, 1.00)	0.2664	0.0596	4.0106	5.7074	0.2886	0.2429	5.9616	7.7924	0.3487	0.3241	8.946	11.2066
	(1.00, 1.03)	0.8905	-0.8507	4.3839	6.4518	0.2406	-0.0325	3.0506	4.7468	0.251	0.1779	4.7421	6.5059
	(1.03, 1.06)	2.2551	-2.2163	6.8082	9.2577	0.6022	-0.5426	3.8383	6.0644	0.2681	-0.0823	3.2398	5.3104
	>1.06	5.6392	-5.585	12.3458	16.5811	3.2206	-3.155	8.9246	13.1826	2.0698	-1.9962	6.9678	10.7667

Panel B: Prediction errors for the financial crisis period from 2007 to 2009 for econometric models

Model	moneyness	< 30 MAPE	< 30 MPE	< 30 MAE	< 30 RMSE	30 - 60 MAPE	30 - 60 MPE	30 - 60 MAE	30 - 60 RMSE	> 60 MAPE	> 60 MPE	> 60 MAE	> 60 RMSE
CGMY	<0.94	0.0478	-0.0401	2.348	2.8772	0.0606	-0.0545	3.05	3.7305	0.0475	-0.0381	2.4168	3.2088
	(0.94, 0.97)	0.2001	-0.1991	5.4824	6.0793	0.264	-0.2637	7.5051	8.0268	0.2312	-0.2304	6.8419	7.8259
	(0.97, 1.00)	0.4768	-0.4752	5.5879	6.1943	0.6374	-0.6374	9.1639	9.6138	0.6055	-0.6052	9.7253	10.5746
	(1.00, 1.03)	0.4985	-0.275	2.4455	3.3785	0.8103	-0.8016	6.1835	7.0284	0.9153	-0.9152	8.4825	9.2707
	(1.03, 1.06)	0.4531	-0.1396	1.3787	1.9559	0.5078	-0.3946	2.6773	3.8832	0.8373	-0.8165	4.6521	5.5418
	>1.06	0.6333	-0.4583	1.0332	1.2684	0.563	-0.43	1.5164	2.207	0.5179	-0.3876	1.8096	2.7381
LSM-BS	<0.94	0.136	0.1358	7.982	9.2863	0.2883	0.288	18.6461	20.1556	0.3968	0.3968	29.0514	30.5315
	(0.94, 0.97)	0.2635	0.2553	7.9583	9.4741	0.4867	0.4865	16.4949	17.9	0.685	0.685	26.6524	27.5667
	(0.97, 1.00)	0.3126	0.2043	5.0572	6.6129	0.5604	0.5575	12.0006	13.5228	0.7424	0.7424	19.3242	20.5624
	(1.00, 1.03)	0.5676	-0.0765	3.6204	5.018	0.619	0.607	8.7453	10.484	0.7841	0.7841	14.5346	15.9967
	(1.03, 1.06)	0.7247	-0.0665	2.9108	4.2952	0.6956	0.6555	6.4242	8.1742	0.8311	0.8311	11.0045	12.4943
	>1.06	0.8015	0.449	2.5138	3.6773	0.836	0.7951	3.8144	5.2348	0.9186	0.9178	5.7087	7.2766
LSM-GARCH	<0.94	0.0357	0.0151	1.9828	3.0248	0.0729	0.068	4.2399	6.1279	0.095	0.0937	6.1814	8.1572
	(0.94, 0.97)	0.0961	0.0432	2.8235	4.3875	0.1871	0.1716	6.3893	8.6709	0.2697	0.2648	10.7346	12.8781
	(0.97, 1.00)	0.2328	0.0604	3.4714	5.034	0.3051	0.227	6.7733	8.6481	0.3725	0.3505	10.5376	12.6393
	(1.00, 1.03)	0.5086	0.192	3.7465	5.2361	0.4595	0.3207	6.7127	8.6096	0.4898	0.466	10.1455	12.3182
	(1.03, 1.06)	0.738	0.7067	3.6716	5.1284	0.6251	0.5607	6.3204	8.2126	0.6534	0.6457	9.5791	11.3947
	>1.06	0.977	0.977	3.1141	4.1834	0.9204	0.9186	4.1727	5.5484	0.8975	0.8968	5.6458	7.2143
LSM-Heston	<0.94	0.1409	-0.1353	8.1849	9.3745	0.0928	-0.0456	5.8104	7.4116	0.0887	-0.0309	6.0148	7.5943
	(0.94, 0.97)	0.3241	-0.2979	9.2181	10.6649	0.2086	-0.0541	6.9365	8.6535	0.1914	0.0216	7.4981	9.8093
	(0.97, 1.00)	0.7266	-0.6644	8.6923	10.2889	0.3432	-0.0665	6.5836	8.463	0.3314	0.1257	8.5627	10.7263
	(1.00, 1.03)	2.0616	-1.9763	8.07	9.4868	0.5733	-0.228	6.0203	7.848	0.4664	0.0995	8.3005	10.444
	(1.03, 1.06)	2.2678	-2.1512	5.8804	6.8342	0.8734	-0.5037	5.0691	6.7679	0.5946	0.0523	7.0568	8.947
	>1.06	1.4007	-0.894	3.0889	4.1501	0.817	-0.0443	3.166	4.5825	0.752	0.2703	4.2538	5.9663
B-W	<0.94	0.0534	0.0514	2.8977	3.8804	0.0892	0.0878	5.0752	6.8473	0.1024	0.1012	6.5073	8.5541
	(0.94, 0.97)	0.1357	0.135	3.9234	5.4925	0.2465	0.2453	8.3134	10.2455	0.3186	0.3185	12.5489	14.5038
	(0.97, 1.00)	0.2467	0.202	4.1248	5.7427	0.4264	0.4264	9.3965	10.9857	0.555	0.555	14.833	16.2828
	(1.00, 1.03)	0.4464	0.1719	3.5186	5.0938	0.5954	0.5954	8.5932	10.2403	0.7157	0.7157	13.5318	15.0098
	(1.03, 1.06)	0.5511	0.3294	2.8593	4.4379	0.7092	0.7086	6.726	8.4302	0.8204	0.8204	11.0141	12.4628
	>1.06	0.7421	0.6895	2.5567	3.7712	0.8649	0.8639	3.9794	5.3823	0.9355	0.9355	5.8165	7.3559
AH-BS	<0.94	0.0655	0.0162	3.8182	4.7686	0.0609	0.017	3.9925	5.0201	0.0688	0.0472	5.073	6.171
	(0.94, 0.97)	0.1242	0.0175	3.4711	4.3887	0.117	0.031	3.7331	4.7782	0.1051	0.0516	3.9414	5.0657
	(0.97, 1.00)	0.253	-0.0048	3.3985	4.2337	0.1795	0.041	3.452	4.3378	0.1514	0.0708	3.6396	4.5117
	(1.00, 1.03)	0.5777	-0.2113	3.0466	3.8115	0.2859	0.016	3.1724	3.9825	0.2185	0.0772	3.4539	4.3342
	(1.03, 1.06)	0.8524	-0.4608	2.6925	3.5403	0.4634	-0.0821	3.0538	3.8348	0.3077	0.0921	3.2345	4.0863
	>1.06	1.2075	-0.8682	2.5964	3.4229	0.8734	-0.4593	2.5903	3.3568	0.6823	-0.2249	2.7908	3.5877
AH-LV	<0.94	0.0643	0.0332	3.7724	4.7171	0.0624	0.034	4.0763	5.1329	0.0961	0.0839	7.2883	14.1939
	(0.94, 0.97)	0.1183	0.028	3.3321	4.2696	0.1161	0.0407	3.7525	4.7054	0.1383	0.1031	5.4253	8.3998
	(0.97, 1.00)	0.2428	0.0457	3.2965	4.1347	0.1852	0.0467	3.5676	4.5302	0.1991	0.1216	5.0538	7.5766
	(1.00, 1.03)	0.5604	-0.1619	3.0389	3.8121	0.3021	0.0433	3.4247	4.2849	0.2594	0.1313	4.3364	5.9526
	(1.03, 1.06)	0.8402	-0.3854	2.6914	3.4782	0.4741	-0.0403	3.1743	4.0351	0.3292	0.0953	3.6837	4.7205
	>1.06	1.1815	-0.7903	2.5477	3.3581	0.8744	-0.4208	2.6647	3.4786	0.685	-0.1787	2.9749	3.9283

Panel C: Prediction errors for the post-crisis period from 2010 to 2012 for machine learning models

Model	moneyness	Days to Expiration											
		< 30				30 - 60				> 60			
		MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
GBNN	<0.94	0.0299	0.0036	1.8615	3.9739	0.0472	0.0286	2.7819	4.3109	0.1053	0.0931	8.0935	12.5593
	(0.94, 0.97)	0.0323	0.0168	0.8808	1.1621	0.0477	0.0349	1.4614	2.091	0.1165	0.1147	4.1002	4.7582
	(0.97, 1.00)	0.0656	-0.0096	0.7855	1.093	0.072	0.0154	1.3177	1.9156	0.1045	0.0898	2.4768	3.2747
	(1.00, 1.03)	0.1846	-0.0169	0.8689	1.2495	0.1163	0.0235	1.2884	1.9683	0.1222	0.0875	2.0086	2.7043
	(1.03, 1.06)	0.2577	0.0471	0.7543	1.1997	0.1734	0.0264	1.1608	1.9208	0.1753	0.1348	2.0661	2.8481
	>1.06	0.314	-0.0163	0.6736	1.0056	0.2278	0.0092	0.7437	1.3136	0.2905	-0.0294	1.2273	2.0868
BNN	<0.94	0.118	0.1157	7.3885	15.4038	0.0931	0.0348	5.8549	10.9107	0.1749	0.0276	12.7616	19.9972
	(0.94, 0.97)	0.1004	0.0714	2.7425	3.8495	0.0682	0.0193	2.0462	2.9256	0.1231	-0.1005	4.3174	5.992
	(0.97, 1.00)	0.1509	-0.0921	1.8073	2.3391	0.059	-0.0138	1.1387	1.5728	0.1017	-0.0617	2.4574	3.6617
	(1.00, 1.03)	0.6042	-0.5931	2.4886	2.9551	0.0869	0.0012	0.9891	1.3721	0.115	0.0155	2.0077	2.7918
	(1.03, 1.06)	0.8846	-0.872	2.2732	2.7132	0.1562	0.047	1.1072	1.4861	0.2023	0.173	2.3718	2.8885
	>1.06	0.9427	-0.9326	1.7721	2.1323	0.2793	-0.0604	0.8724	1.2442	0.3432	0.2429	1.9141	2.5009
SVR	<0.94	0.2445	0.2048	13.5636	19.973	0.1545	-0.0143	9.7602	17.6914	0.5837	-0.5765	47.4798	77.7271
	(0.94, 0.97)	0.3352	0.3299	9.2463	10.5165	0.1681	0.1188	4.9959	6.1755	0.2492	-0.2255	8.9135	12.7452
	(0.97, 1.00)	0.2752	0.1918	3.696	4.6771	0.1687	0.0909	3.0279	3.7969	0.2393	-0.2129	5.7309	7.7936
	(1.00, 1.03)	0.6247	-0.3683	2.4504	3.08	0.2097	0.0225	2.2137	2.7805	0.2398	-0.175	3.9703	5.4472
	(1.03, 1.06)	1.0331	-0.7746	2.3432	2.9819	0.3735	-0.0779	2.148	2.6417	0.2483	-0.1216	2.7941	3.6498
	>1.06	1.0646	-0.7729	1.9508	2.5479	0.6979	-0.2682	1.7646	2.2691	0.5519	-0.1484	1.992	2.6075
GP	<0.94	0.4977	0.479	28.8643	38.7371	0.5236	0.5154	30.5782	39.4294	0.5482	0.5405	39.7942	51.2143
	(0.94, 0.97)	0.4217	0.4129	11.8527	14.6984	0.4483	0.443	13.7194	16.6998	0.4211	0.4159	15.0355	18.6636
	(0.97, 1.00)	0.2645	0.1176	3.6762	5.085	0.3329	0.318	6.4074	8.14	0.3921	0.3799	9.2505	11.2426
	(1.00, 1.03)	1.036	-0.9925	3.634	4.9763	0.2295	0.04	2.6114	3.6782	0.2792	0.2463	4.7592	6.1927
	(1.03, 1.06)	2.3722	-2.3373	5.3304	6.7841	0.57	-0.5032	2.9788	4.1665	0.2065	0.047	2.4371	3.5193
	>1.06	4.1984	-4.1393	7.7413	10.0433	2.5863	-2.5325	5.6927	7.5084	1.8182	-1.7517	4.7925	6.6273

Panel C: Prediction errors for the post-crisis period from 2010 to 2012 for Econometric models

Model	moneyness	< 30 MAPE	< 30 MPE	< 30 MAE	< 30 RMSE	30 - 60 MAPE	30 - 60 MPE	30 - 60 MAE	30 - 60 RMSE	> 60 MAPE	> 60 MPE	> 60 MAE	> 60 RMSE
CGMY	<0.94	0.0355	0.0355	1.6113	1.6874	0.049	0.049	2.4566	2.5736	0.059	0.059	3.7519	3.9107
	(0.94, 0.97)	0.0513	0.0335	1.3466	1.5393	0.0882	0.053	2.4482	2.8341	0.1576	0.157	5.1994	5.9468
	(0.97, 1.00)	0.1983	-0.0376	2.1841	2.524	0.2033	0.0354	3.3061	3.6756	0.2059	0.1309	4.2657	4.9837
	(1.00, 1.03)	0.6256	-0.2681	2.5831	3.0448	0.3643	-0.0141	3.4654	3.8416	0.2855	0.1612	4.1644	4.8075
	(1.03, 1.06)	0.867	-0.5297	1.9777	2.4792	0.499	-0.1212	2.6908	3.1028	0.3339	0.0607	3.2137	3.6869
	>1.06	0.8047	-0.5741	1.2633	1.6436	0.5355	-0.3186	1.2926	1.7194	0.3425	0.1064	1.4146	1.9135
LSM-BS	<0.94	0.0799	0.0367	4.0652	4.9764	0.1608	0.1293	8.6239	9.9318	0.2134	0.2073	13.7504	15.077
	(0.94, 0.97)	0.24	-0.1836	6.4173	8.8862	0.2071	0.0157	6.241	8.1326	0.3019	0.1792	10.4758	12.1311
	(0.97, 1.00)	1.0266	-1.0149	11.7044	14.2062	0.4961	-0.3605	8.506	11.6361	0.38	-0.0059	8.695	10.7056
	(1.00, 1.03)	3.5942	-3.5896	13.6059	15.6831	1.0001	-0.9147	9.5216	12.8416	0.5154	-0.1669	8.1398	10.7476
	(1.03, 1.06)	5.2542	-5.2473	11.1444	13.1047	1.7797	-1.6904	8.7624	11.736	0.6372	-0.2602	6.7725	9.2245
	>1.06	3.435	-3.3195	5.8067	7.8841	2.2465	-1.9739	5.167	7.618	1.1302	-0.5015	4.2368	6.3526
LSM-GARCH	<0.94	0.0204	-0.0132	1.0278	1.2537	0.0355	0.0303	1.765	2.4548	0.0722	0.0722	4.1189	4.8879
	(0.94, 0.97)	0.0554	-0.0129	1.5228	2.0064	0.0946	0.0843	2.9805	4.2644	0.2047	0.2045	7.3237	8.4027
	(0.97, 1.00)	0.1568	-0.0427	1.9501	2.6447	0.1632	0.1142	3.271	4.7141	0.2204	0.2042	5.4817	6.9661
	(1.00, 1.03)	0.3171	-0.0023	1.6954	2.5749	0.2698	0.1876	3.3188	4.7085	0.3615	0.3579	6.2576	7.6325
	(1.03, 1.06)	0.5638	0.5343	1.9283	2.8211	0.4818	0.4665	3.6317	4.8764	0.555	0.555	6.6406	7.73
	>1.06	0.9439	0.9439	2.2239	2.7294	0.8521	0.852	2.784	3.5707	0.843	0.843	4.0628	5.1286
LSM-Heston	<0.94	0.1409	-0.1353	8.1849	9.3745	0.0928	-0.0456	5.8104	7.4116	0.0887	-0.0309	6.0148	7.5943
	(0.94, 0.97)	0.3241	-0.2979	9.2181	10.6649	0.2086	-0.0541	6.9365	8.6535	0.1914	0.0216	7.4981	9.8093
	(0.97, 1.00)	0.7266	-0.6644	8.6923	10.2889	0.3432	-0.0665	6.5836	8.463	0.3314	0.1257	8.5627	10.7263
	(1.00, 1.03)	2.0616	-1.9763	8.07	9.4868	0.5733	-0.228	6.0203	7.848	0.4664	0.0995	8.3005	10.444
	(1.03, 1.06)	2.2678	-2.1512	5.8804	6.8342	0.8734	-0.5037	5.0691	6.7679	0.5946	0.0523	7.0568	8.947
	>1.06	1.4007	-0.894	3.0889	4.1501	0.817	-0.0443	3.166	4.5825	0.752	0.2703	4.2538	5.9663
B-W	<0.94	0.039	0.0335	1.9644	2.1548	0.0696	0.0619	3.4885	3.927	0.0889	0.0889	5.1729	5.7899
	(0.94, 0.97)	0.1281	-0.0664	3.4179	4.809	0.1428	0.0556	4.3577	5.4615	0.2227	0.1728	7.8271	8.7966
	(0.97, 1.00)	0.6738	-0.6511	7.5474	9.4919	0.3251	-0.163	5.7123	7.8078	0.3011	0.1004	7.1006	8.2234
	(1.00, 1.03)	2.4966	-2.4795	9.2033	10.932	0.6472	-0.4997	6.2856	8.6468	0.4085	0.0386	6.6667	7.9261
	(1.03, 1.06)	3.4306	-3.4065	7.0435	8.7523	1.1091	-0.9345	5.5565	7.741	0.5077	0.041	5.754	7.046
	>1.06	1.8801	-1.7029	3.0989	4.6276	1.2669	-0.8702	3.0559	4.6719	0.7924	0.0426	3.4901	4.6995
AH-BS	<0.94	0.0661	0.023	3.3814	4.1424	0.0744	0.0393	3.8424	4.8128	0.0687	0.0485	4.4384	5.5515
	(0.94, 0.97)	0.1345	0.0332	3.596	4.4328	0.1209	0.0418	3.5793	4.4968	0.1127	0.063	3.9144	4.6158
	(0.97, 1.00)	0.267	-0.0064	3.2467	4.0543	0.1936	0.0424	3.4175	4.2608	0.1723	0.1001	3.9493	4.8428
	(1.00, 1.03)	0.6483	-0.264	2.8363	3.6067	0.3143	0.0599	3.2914	4.0397	0.2277	0.1225	3.6426	4.5899
	(1.03, 1.06)	1.0015	-0.6317	2.4553	3.2222	0.482	-0.0274	2.8559	3.5649	0.305	0.1457	3.3113	4.186
	>1.06	1.2987	-0.9443	2.3985	3.2311	1.0029	-0.5858	2.4503	3.2241	0.7453	-0.2712	2.5962	3.2886
AH-LV	<0.94	0.0672	0.0237	3.4705	4.3512	0.0788	0.0463	4.0259	5.0545	0.1712	0.1493	12.2149	25.3507
	(0.94, 0.97)	0.1242	0.0318	3.3288	4.1655	0.1259	0.0434	3.7453	4.8152	0.1605	0.1113	5.6667	8.1279
	(0.97, 1.00)	0.2715	0.0271	3.3239	4.1098	0.1891	0.0503	3.3713	4.2481	0.1947	0.1131	4.5605	6.53
	(1.00, 1.03)	0.6115	-0.1579	2.704	3.4187	0.3159	0.0697	3.338	4.1464	0.2645	0.1699	4.3267	5.7998
	(1.03, 1.06)	1.0453	-0.6348	2.5004	3.2811	0.4869	-0.0116	2.9683	3.7447	0.3737	0.204	4.0854	5.1047
	>1.06	1.2679	-0.9048	2.3237	3.0622	0.9727	-0.5406	2.4165	3.1603	0.78	-0.269	2.8048	3.6704

(Column groups: Days to Expiration < 30, 30 - 60, > 60)

Table 5.7 7-day ahead Prediction performance. Table reports the prediction error results for S&P 100 index American put options of each categories with respect to the moneyness, κ, and time to maturity, τ. Abbreviations as in Table 3.8.

Panel A:Prediction errors for pre-crisis period from 2003 to 2006.

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
GBNN	<0.94	0.0758	-0.0018	4.3824	9.6444	0.0677	0.0272	4.1684	9.2566	0.0683	0.0617	3.8903	6.3409
	(0.94, 0.97)	0.0394	-0.0076	0.9511	1.3094	0.04	0.0114	1.0562	1.4266	0.0734	0.0641	2.1901	2.9441
	(0.97, 1.00)	0.0858	-0.0473	0.8105	1.0185	0.071	-0.0222	0.9767	1.3321	0.088	0.0323	1.5975	2.3232
	(1.00, 1.03)	0.2755	-0.1399	0.8509	1.068	0.151	-0.0139	1.0127	1.3575	0.1285	0.0437	1.3463	1.9677
	(1.03, 1.06)	0.3887	-0.2368	0.7573	0.9453	0.2409	-0.1024	0.835	1.1691	0.1911	0.0557	1.198	1.7956
	>1.06	0.3776	-0.2047	0.6123	0.7829	0.2859	-0.1328	0.6296	0.8916	0.292	-0.0985	0.7902	1.3028
CGMY	<0.94	0.0873	-0.0774	3.4656	4.341	0.114	-0.0926	4.869	6.4774	0.105	-0.0672	4.8782	6.6633
	(0.94, 0.97)	0.1639	-0.1452	3.8443	5.4151	0.2026	-0.1782	5.2789	7.788	0.1967	-0.1482	5.748	8.5647
	(0.97, 1.00)	0.381	-0.3768	3.8723	5.0494	0.4135	-0.4078	5.6987	7.8216	0.3209	-0.2877	5.6541	8.3838
	(1.00, 1.03)	0.8589	-0.8271	2.9887	3.9555	0.7215	-0.7088	5.1336	6.9696	0.5115	-0.4865	5.5194	8.2128
	(1.03, 1.06)	1.1896	-1.1319	2.3462	3.0913	0.9608	-0.9371	3.5999	5.2037	0.703	-0.6696	4.6632	7.1189
	>1.06	0.9492	-0.8488	1.3878	1.9621	1.1941	-1.1559	2.6053	3.7058	0.9826	-0.9417	3.2114	5.1093

Panel B: Prediction errors for the financial crisis period from 2007 to 2009.

		Days to Expiration											
		< 30				30 - 60				> 60			
Model	moneyness	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE	MAPE	MPE	MAE	RMSE
GBNN	<0.94	0.072	-0.013	6.4264	11.3972	0.0721	-0.0007	5.3598	12.0626	0.0928	0.0105	7.6781	14.9486
	(0.94, 0.97)	0.0847	-0.029	2.4809	7.3793	0.1058	-0.0503	3.4432	7.7284	0.1285	-0.0746	4.6911	9.798
	(0.97, 1.00)	0.1532	-0.0751	2.2265	3.1361	0.1832	-0.0882	3.6282	3.9328	0.2208	-0.1135	5.1973	6.2377
	(1.00, 1.03)	0.3325	-0.1549	2.1712	3.1018	0.2729	-0.1112	3.4189	3.7227	0.3259	-0.1827	5.2785	6.3283
	(1.03, 1.06)	0.4097	-0.1235	1.8488	2.9767	0.372	-0.1784	2.9604	4.3216	0.4396	-0.2619	4.7822	5.8082
	>1.06	0.533	-0.1569	1.5319	3.5914	0.4127	-0.1699	1.6063	2.7585	0.4485	-0.2248	2.1643	3.9102
CGMY	<0.94	0.0506	0.0262	2.7721	3.499	0.08	0.0528	4.622	5.8964	0.0817	0.0617	5.3621	7.0513
	(0.94, 0.97)	0.2493	-0.0608	4.2903	7.1855	0.2991	-0.0378	6.4977	9.5361	0.2797	-0.0061	6.8023	8.3615
	(0.97, 1.00)	0.457	-0.2458	4.9261	7.9187	0.7069	-0.2737	7.4039	10.5934	0.7422	-0.1735	7.6694	11.0531
	(1.00, 1.03)	0.5448	-0.2587	3.372	5.1307	0.8055	-0.4272	6.2555	7.478	0.9108	-0.3213	7.4115	10.7638
	(1.03, 1.06)	0.596	0.0052	2.4209	3.8059	0.6881	-0.4057	4.6048	5.8367	0.8068	-0.364	5.8767	8.0877
	>1.06	0.6996	0.3377	2.1419	3.3432	0.5914	0.0607	2.676	5.0567	0.572	-0.0036	3.3217	6.7227

Panel C: Prediction errors for the post-crisis period from 2010 to 2012.

Model	moneyness	< 30 MAPE	< 30 MPE	< 30 MAE	< 30 RMSE	30 - 60 MAPE	30 - 60 MPE	30 - 60 MAE	30 - 60 RMSE	> 60 MAPE	> 60 MPE	> 60 MAE	> 60 RMSE
GBNN	<0.94	0.0444	0.0062	3.2486	8.1772	0.0572	0.0411	3.8624	7.89	0.1316	0.1227	10.678	17.5844
	(0.94, 0.97)	0.0381	0.0162	1.0461	1.3932	0.0582	0.0372	1.8759	2.705	0.1262	0.1239	4.7329	5.5818
	(0.97, 1.00)	0.0918	-0.0187	1.2054	1.788	0.0919	0.0182	1.8685	2.7991	0.1237	0.095	3.2643	4.5039
	(1.00, 1.03)	0.2403	-0.0583	1.3282	2.0111	0.1473	0.0143	1.8788	2.8905	0.1476	0.0979	2.8136	4.0869
	(1.03, 1.06)	0.3485	-0.0288	1.2408	1.9739	0.2228	0.0036	1.7816	2.8061	0.2058	0.1178	2.8679	4.204
	>1.06	0.4271	-0.0456	1.052	1.5385	0.3468	-0.0739	1.2	2.0039	0.3924	-0.0968	1.7383	2.9523
CGMY	<0.94	0.0399	0.0399	1.9952	2.1845	0.072	0.072	3.6	4.0457	0.0959	0.0959	5.5261	6.1711
	(0.94, 0.97)	0.1005	0.1005	2.7112	3.2791	0.1914	0.1914	5.8328	6.6813	0.3035	0.3035	10.7454	11.4528
	(0.97, 1.00)	0.1995	0.1792	2.7677	3.8004	0.3139	0.3137	6.0847	7.2383	0.4159	0.4159	10.0153	10.9285
	(1.00, 1.03)	0.7544	0.1177	1.9829	3.0993	0.366	0.3546	4.5376	5.9119	0.4718	0.4718	8.0726	9.161
	(1.03, 1.06)	0.8261	0.1703	1.5626	2.5551	0.5974	0.3267	3.1424	4.521	0.4973	0.4973	6.1311	7.3576
	>1.06	0.8968	0.4193	1.4002	2.1049	0.6223	0.3169	1.7234	4.7733	0.4707	0.4467	2.8531	5.2393

국문초록

본 논문은 전통적인 계량 경제 모형 및 기계 학습 모형을 활용하여 금융 파생상품 시장 및 암호화폐시장 분석을 수행하고자 하였다. 주어진 시장 데이터를 활용한 체계적인 경험적 실험 결과를 통해 시장 설명 능력 및 시장 예측 능력을 기반으로 기존 연구에서 다양하게 활용되어온 계량 경제 모형 및 기계 학습 모형을 평가하여 각 모형의 한계를 파악하고 설명 및 예측 능력 향상을 위한 데이터 기반 기계 학습 방법론을 제안하였다.

GARCH 모형 및 확률 변동성 모형과 같은 계량 경제 모형 내에 은닉 변수가 관찰 변수와 개별적으로 시계열 모형을 구성하는 경우, 기존 모형의 모수 추정 방법론은 느린 수렴 속도 및 빈번한 지역해 문제가 발생한다. 특히 일반적인 MCMC 방법론의 경우 구체적인 목표확률분포를 알 수 없지만 그와 비슷하거나 확률적으로 가까운 후보확률분포를 선택하는 것이 전체 성능을 결정하는 중요한 요소이다 본 연구에서는 generative model이라는 개념을 사용하여 후보확률분포를 정확하게 제안하는 대신, Kullback-Leibler (KL) distance 관점에서 목표확률분포와 가까운 후보확률분포로부터 추출된 다량의 샘플을 제공함으로써 MCMC 방법론의 성능을 향상시키고자 하였다. Generative 모형으로부터 샘플을 얻기 때문에 매우 단시간에 샘플을 추출하는 것이 가능하고 구체적인 확률분포를 제안하는 대신 KL divergence 관점에서 가까운 확률분포로부터 얻은 샘플만을 제공함으로써 기존의 후보확률분포에 의존적인 MCMC 방법론의 단점을 개선하였다.

또한 2012년의 S&P 100 인덱스 옵션데이터를 활용하여 대표적인 점프 발산 모형인 CGMY, Kou모형과 대표적인 기계학습 모형인 인공신경망, 베이지안 인공신경망, 서포트 벡터머신, 가우시안 프로세스 모형들에 대하여 in-sample, out-of-sample의 에러를 측정하여 모델 추정 및 예측 능력을 비교하였다. 모델 추정의 경우, 점프 발산 모형은 최근 하루의 데이터를 사용하여 모델을 추정한 성능이 제일 좋은 반면 기계학습 모형은 최근 일주일 간의 데이터를 활용한 모델 추정 성능이 제일 높은 것으로 나왔다. 반면 예측 성능의 경우, 점프 발산 모형 및 베이지안 인공신경망의 성능이 비슷하게 가장 좋았으며, 다른 기계학습의 경우 예측하는 범위가 넓어질수록 성능이 빠르게 떨어지는 것을 확인하였다. 특히 본 연구에서는 점프 발

산 모형이 아메리칸 옵션 및 유러피언 옵션 간의 도메인 적응 측면에서 매우 높은 성능을 보이는 것을 확인하였으며 이러한 차이는 점프 발산 모형이 아메리칸 옵션 및 유러피언 옵션이 공통된 자산을 기반으로 한다는 사실을 모형 내에 반영하고 있기 때문이라고 예상 할 수 있다. 이러한 경험적 선행 연구를 바탕으로, 생성적 베이지안 인공신경망 (GBNN) 이라는 모델을 제안함으로써 기계 학습 모형이 가지고 있는 단점을 극복하고자 하였다. 일반적인 기계학습 방법론은 데이터로부터 모형을 학습하기 때문에 옵션 시장에 관한 모형을 학습시킬 때 데이터가 없는 깊은 ITM 이나 깊은 OTM영역에서 성능이 매우 빠르게 감소한다. GBNN은 이 문제를 해결하기 위해 어떤 영역에서도 적절한 가격을 제안하는 임의의 발산모형으로부터 얻어진 가상의 가격데이터를 GBNN의 초기 학습 시에 추가하였다. 다음 학습 시기가 왔을 때, GBNN은 전 날까지 학습된 GBNN의 데이터로부터 사전 확률에 대한 정보를 얻으며 학습 날의 실제 거래 데이터로부터 우도확률 값을 얻어 사후 확률을 최대로 하는 모형을 학습하였다. 그 결과, 2003년부터 2012년 까지 S&P 100 인덱스 아메리칸 옵션 데이터에 대해 일반적인 기계학습 모형과는 달리 GBNN의 깊은 ITM 및 OTM 영역의 추정 및 예측 성능이 눈에 띄게 향상하였고 점프 확산 모형보다 매우 좋거나 비슷한 추정 및 예측 성능을 보였다. 특히 선행 연구에서 모형 적합 시 적용 데이터 범위에 따라 모형 적합 성능이 매우 변동성이 큰 일반적인 기계학습 모형과는 달리 GBNN은 모형 적합 성능이 다른 방법론들에 비하여 매우 빠르고 안정적인 것을 확인할 수 있었다. 특히 독립된 50회의 모형 적합을 수행한 결과 매 모형 적합 시 결과 성능의 변동성이 매우 큰 점프 발산 모형에 비하여 GBNN의 경우 매 적합마다 일정한 성능 범위 안에 포함되는 것을 확인할 수 있었다. 또한 적합된 모형에 대해 옵션 가격 계산 시간 측면에서 GBNN이 월등히 빠른 것을 확인할 수 있었는데 이는 모형이 형성된 뒤에도 얻어진 모수를 기반으로 자산의 예측 값을 먼저 계산하고 주어진 가격 결정 방법을 통해 옵션 가격을 결정하는 계량 경제학 모형과는 달리 모형이 형성되고 난 뒤에는 테스트 측면에서 매우 빠른 속도를 보이는 인공 신경망의 특징을 반영한 것으로 보인다.

2008년 사토시 나카모토가 제안한 분산 원장이라는 개념을 기술적으로 구현한 최초의 암호화페인 비트코인을 필두로 하여 다양한 종류의 암호화페가 개발되면서 최근의 암호화페시장이 형성되었다. 암호화페시장은 기존에 없던 새로운 고유한 특징을 가지는 시장으로써 전통적인 시장 분석 기술뿐만 아니라 암호 화페시장에 적합한 새로운 분석 기술에 관한 수요가 증가하고 있는 상황이다. 본 연구에서는

기존의 전통적인 계량 경제학 방법론에 기반을 두는 선행 연구에서 나아가 분산 원장을 기술적으로 구현한 블록체인의 정량적인 데이터를 활용하여 대표적인 암호화폐인 비트코인 시계열을 분석하였다. 블록체인의 데이터를 고려한 베이지안 인공 신경망은 다른 벤치마크 모형들에 비해 높은 예측 성능 및 추정 성능을 보였으며 선행 연구들에 비해 최근의 암호 화폐의 큰 변동성을 반영하는 것을 확인하였다. 비트코인으로부터 파생되는 많은 알트코인이 기술적으로는 같은 코드로부터 개발되었기 때문에 암호 화폐들 간의 가격 사이에 상관관계가 있을 것이라는 가정 하에 확장된 GRU 모형을 사용하여 화폐 간의 상관관계 분석을 수행하였다. 전통적인 시장 모형인 Vector Autoregressive (VAR)모형이 다변수간의 상관관계를 선형 모형으로 가정하는 것에 착안하여 GRU 모형으로부터 구해진 게이트 값을 VAR 모형의 모수로 가정하고 기존에 추정에 어려움이 있었던 매 시간의 충격파 간의 공분산 행렬을 인공신경망을 통해 추정함으로써 암호화폐 시장의 다양한 대안화폐들 간의 상관관계의 시각화를 가능하게 하였다. 그 결과 대안 화폐들 간의 긴밀한 상관관계가 존재한다는 것을 확인하였으며 특히 기존의 화폐로부터 분리되어 나온 화폐와 기존 화폐간에 매우 유의미한 상관관계가 존재한다는 것을 확인하였다.

본 연구는 파생상품 및 가상화폐 시장분석을 위한 시계열 data-driven 기술을 개발하여 시장의 정량적인 해석에 관한 연구를 진행하였다. 데이터 기반의 학습 프레임 워크에 대한 기존 연구는 의미 분석에 제한적으로 진행되어 왔으나 본 연구는 해석이 용이한 데이터 기반 기계 학습 보형을 개발하여 데이터로부터의 의미를 시각화하고 해석하는데 집중하였다. 본 연구의 연구 결과를 기반으로 하여 최근 형성된 가상화폐 시장의 시계열 데이터 분석에 기여하는 것이 가능하며 나아가 가상화폐 시장의 파생상품 시장 형성 시에 본 연구에서 다룬 파생상품의 시계열 분석 프레임 워크를 활용하여 확장 적용이 가능할 것으로 예상된다.

주요어: 금융시장분석, MCMC, 암호화폐시장, 기계학습, 베이지안 인공신경망, 시계열 모형 분석
학번: 2015-30239