Accepted Manuscript

Machine learning versus econometric jump models in predictability and domain adaptability of index options

H. Jang, J. Lee

PII: S0378-4371(18)31033-1
DOI: https://doi.org/10.1016/j.physa.2018.08.091
Reference: PHYSA 19977

To appear in: *Physica A*

Received date : 30 January 2018
Revised date : 28 April 2018

Highlights

>Empirical comparison study on the econometric and machine learning models.
>Domain adaptation over the S&P 100 European/ American index options.
>Better prediction performance of econometric models over machine learning models.

Machine Learning versus Econometric Jump Models in Predictability and Domain Adaptability of Index Options

H. Jang[a], J. Lee[a,*]

[a]*Department of Industrial Engineering*
Seoul National University, Seoul National University, San 56-1, Sillim-dong, Gwanak-gu, Seoul 151-742, Republic of Korea.

Abstract

Econometric jump models dealing with key stylized facts in financial option markets have an explicit underlying asset process based on stochastic differential equations. Machine learning models with improved prediction accuracy have elicited considerable attention from researchers in the field of financial application. 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.

Keywords: Financial time series, Lévy process, Bayesian neural network, Neural network, Support Vector Regression, Gaussian process regression, Domain Adaptation

1. Introduction

Machine learning models, which are equipped with outstanding predictability, have elicited considerable attention from many researchers in 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) [1, 2, 3, 4, 5, 6, 7], support vector machines[8, 9], and Gaussian processes (GPs) [10, 11, 12, 13]. These models have also considered different types of available market information, but did not consider explicit formulation for underlying processes.

*Corresponding author. Please address all correspondences to Dr. Jaewook Lee, Department of Industrial Engineering, Seoul National University, San 56-1, Sillim-dong, Gwanak-gu, Seoul 151-742, Republic of Korea. Email: *jaewook@snu.ac.kr*

Email addresses: `gmltn7798@snu.ac.kr` (H. Jang), `jaewook@snu.ac.kr` (J. Lee)

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 [14, 15, 16, 17, 18, 19, 20, 21]. 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.

A considerable number of studies have been conducted to elucidate financial option markets by applying either econometric financial jump models or machine learning models, but not the two models jointly. In this study, an empirical study is conducted to compare econometric financial jump models with state-of-the-art machine learning models in terms of model validity, model predictability, and domain adaptability. 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.

The rest of this paper is organized as follows. Section 2 presents a brief review of machine learning models used in this study. Section 3 introduces econometric financial jump models with the calibration methods for European and American options. Section 4 demonstrates the experiments conducted to answer the preceding questions. The performances of the compared models are evaluated using the S&P 100 Index American/European put options from 2012. Section 5 provides the conclusions.

2. Related work

This section reviews related work of financial derivatives pricing using machine learning and parametric models, and explain explicit conditions for machine learning models covered in this paper. Artificial neural network (ANN) models or recently called deep learning models, consists of several interconnected layers. Layers except for the input and output layer are referred to the hidden layer and each hidden or output layer represents mathematically a non-linear function of the weighted linear combination of the neuron node values in the input or previous hidden layers. The employed non-linear function is referred as an activation function and includes hyperbolic tangent, logistic, or relu function. In financial option markets, various neural networks has been successfully applied to pricing and prediction of financial derivatives. In regard to feedforward neural networks, they include [1] for pricing and delta-hedging of S&P 500 futures options, [2] for daily Australian SPI options data, [4] for Nikkei 225 index future options data, [22] for crude oil, SSE, N225, and DAX, and [23] for FTSE100. Bayesian neural network is another popular class of neural networks proposed to mitigate the over-fitting problem by adding a Bayesian regularization term to the objective function. In regard to Bayesian or hybrid neural networks, they include [5] for pricing and hedging with daily S&P 500 index daily call options and [24] for forecasting the volatility in three Latin-American stock exchange indexes from Brazil, Chile and Mexico. [25] investigate applications of deep learning networks for stock market analysis and prediction. They emphasize the characteristics of deep learning algorithms dependent on the network structure and offer a systematic analysis for several structures.

Successful support vector regression models for predicting financial assets include [8] for the daily market data of AUD/USD, EUR/USD, USD/JPN, and GBP/USD currency options, and [9] for forcasting NASDAQ quotes, namely Intel, National Bank shares, and Microsoft daily closed stock prices. Gaussian process (GP) Regression is an alternative machine learning model that has been attractive due to its ability to predict the distribution of asset prices in recent times. In this research, the "Matérn class" is adopted as covariance functions in our experiments as [10], which is given by

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

where ν and l are positive parameters, and K_ν is a modified Bessel function with ν which controls the degree of smoothness. See [26] for more details on the GP. [10] compared the Gaussian process regression model with NN models for KOSPI 200 call ELWs from March 2006 to July 2006. They shows that the Gaussian process model is superior to three NN models in terms of both pricing and hedging errors. [11] suggested a prediction method for the implied volatility function distribution based on the Gaussian process regression. They shows that the proposed method overcomes the negative price containing problem by the simulation results. [12] proposes the Gaussian process regression methods to predict positive option price distributions and shows the comprehensive empirical study using the KOSPI200 index options.

[27] find that the prediction performance of machine learning models is better than the parametric model such as the Black-Scholes model, the Heston model, and the Merton model by an empirical study based on the KOSPI200 index options from January 2001 to December 2010. [28] compares the predictive accuracy of the financial time series between traditional time series models and mainstream machine learning models including state-of-the-art ones of deep learning using historical real stock index data such as S&P 500, Dow 30, and Nasdaq. The results show machine learning outperforms existing time series models in terms of precision.

Recent study have attempted to propose a new approach considering non-parametric and parametric models to overcome the drawbacks of the market prediction. [13] introduced a new Gaussian process regression framework combined with the stochastic volatility(SV) model and also presented the model estimation method for the Gaussian process regression stochastic volatility model. They analyze the IBM stock daily adjusted closing price data by the proposed framework. [29] proposes a hybrid model by combining nonparametric machine learning models with a parametric option pricing model, such as the Black-Scholes (BS) option pricing model, the Monte Carlo option pricing model and the finite difference method. In addition, They have confirmed that the proposed model is more effective than the benchmark models by investigating an experimental study that predict the one-day-ahead price by using the index option traded on the Indian Stock Exchange. In this paper, we reveals that the parametric models are superior to non-parametric machine learning model in terms of the robustness and domain adaptation ability by an comprehensive empirical study.

3. Econometric financial jump models for option pricing

Econometric financial jump models embody the return process of market variables, including jumps of the return, which can depict the evolution of market variables in the real market [14, 15, 17, 18, 19, 30]. Various numerical methods have been proposed because no closed form solutions exist for option pricing in the case of Lévy jump models. Fast Fourier transform algorithms proposed by [17], [31] and improved by [32, 33] are the most widely used numerical methods for evaluating European call and put options. However, they have not been applied to several intricate derivatives, such as barriers or American-type options in which prices under econometric jump models are represented by solving partial integro-differential equations with several conditions [34]. In this study, the implicit method suggested in [35] is used to preserve the second-order accuracy of time and spatial variables.

3.1. Kou model

The Kou model [36] 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

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

and

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

3.2. CGMY model

the CGMY model [14] is an infinite activity Lévy process model described by the following risk-neutral stock price process

$$\frac{dS_t}{S_t} = (\gamma - q + \omega)dt + dX_t(\nu)$$

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

$$\nu(x) = c\left[\frac{e^{-\lambda_+ x}1_{x>0}}{x^{1+\alpha}} + \frac{e^{-\lambda_-|x|}1_{x<0}}{|x|^{1+\alpha}}\right]$$

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

There are 4 parameters $\theta = (c, \alpha, \lambda_-, \lambda_+)$: c, the frequency of jumps, λ_-, λ_+, the tail behavior of the Lévy measure, and α, the local behavior of the process, which is the gap between big jumps. The characteristic function $\Phi_s(z)$ of $s_t = \ln(S_t)$ has the following form:

$$\Phi_s(z) = \mathbb{E}[e^{izs_t}] = e^t e^{(iz(\gamma-q+\omega)+c\Gamma(-\alpha)((\lambda_+-iz)^\alpha-\lambda_+^\alpha+(\lambda_-+iz)^\alpha-\lambda_-^\alpha)} \tag{3}$$

and

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

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

4. Empirical Studies

4.1. 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 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 [36, 14] 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.

Table 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	

(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 1 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 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.

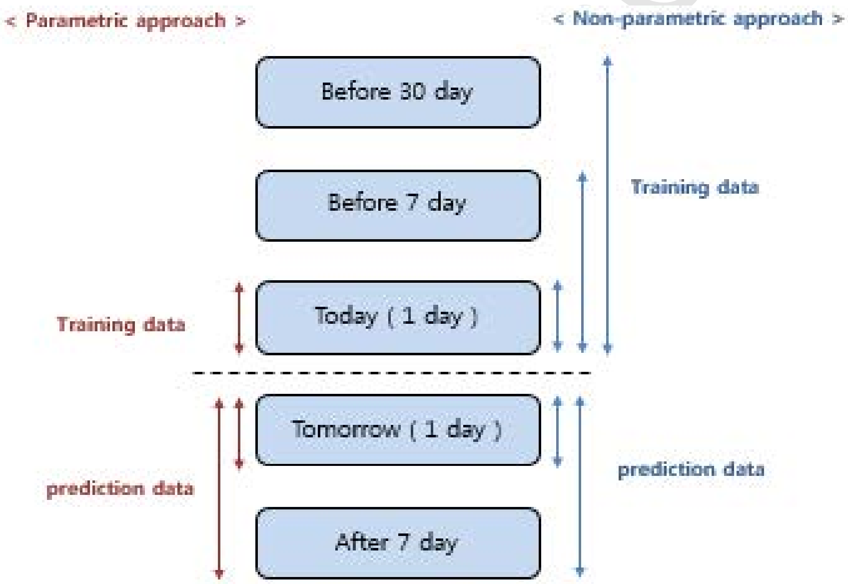

Figure 1: Scheme of experiments for the model estimation and prediction.

4.2. 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 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

Table 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

performance. Although most MPE values in machine learning models are negative (overvalued), they are relatively small in absolute values, thereby indicating unbiased direction similar with econometric jump models.

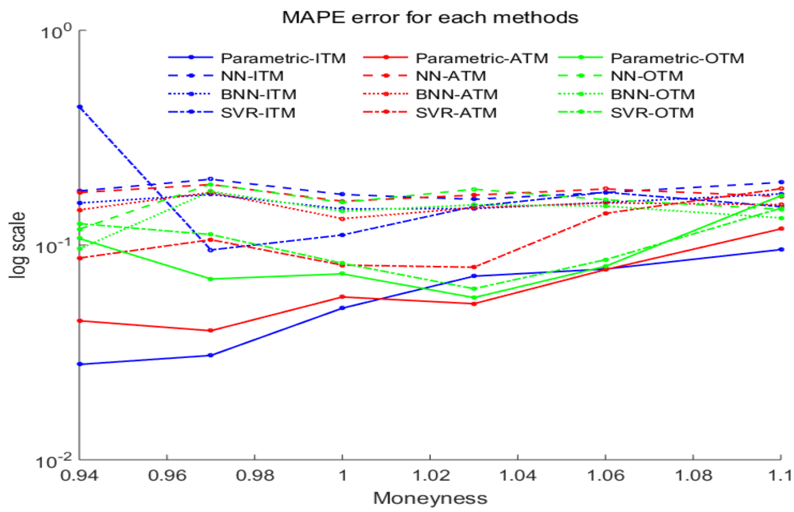

Figure 2: Total MAPE estimation error with respect to moneyness and expiration date.

Figure 2-5 presents detailed results of model estimation errors in terms of moneyness and time to maturity. In-the-money or at-the-money options with short maturity have small estimation errors for econometric jump and machine learning models; the latter presented 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 overfitting in prediction. Moreover, no significant difference is observed for the estimation errors between econometric jump models using data only from previous one-day and

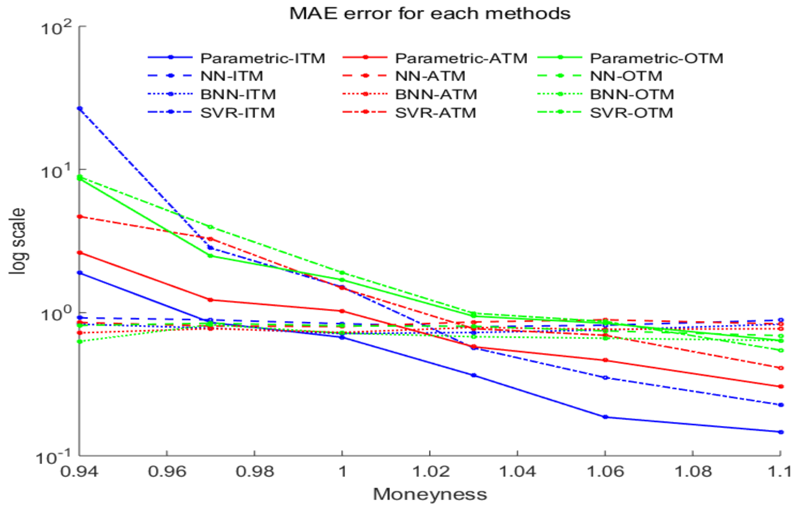

Figure 3: Total MAE estimation error with respect to moneyness and expiration date.

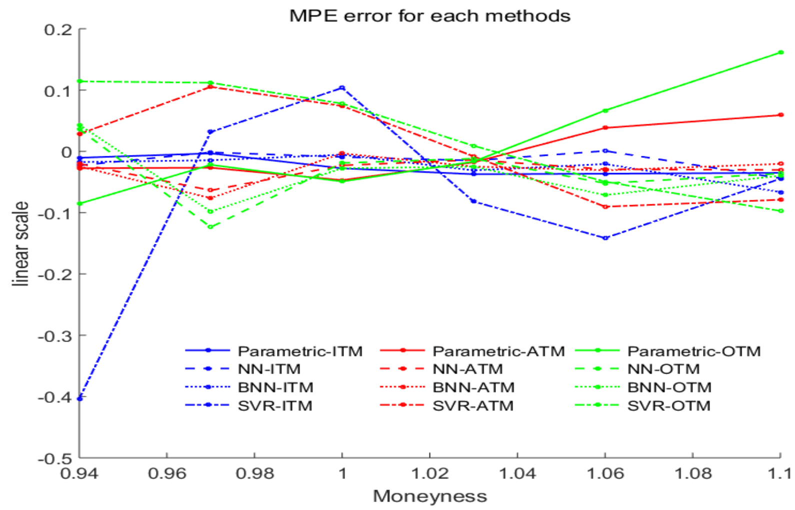

Figure 4: Total MPE estimation error with respect to moneyness and expiration date.

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.

Table 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

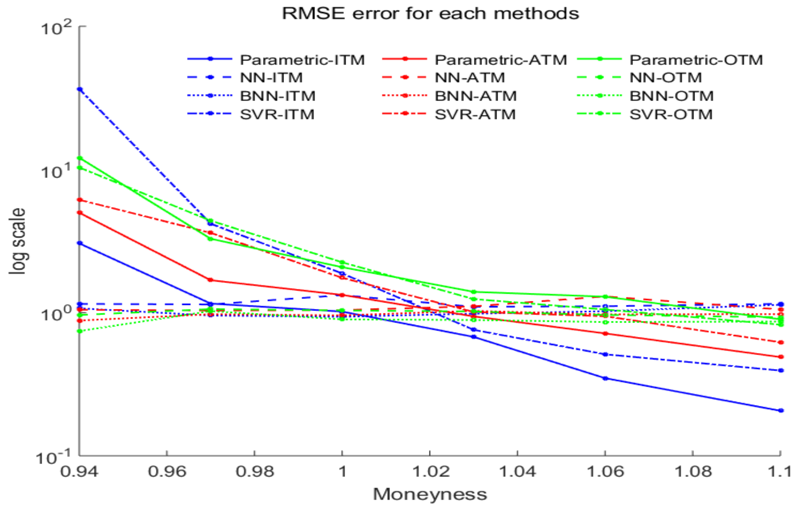

Figure 5: Total RMSE estimation error with respect to moneyness and expiration date.

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 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).

Figure 6-13 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 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 pro-

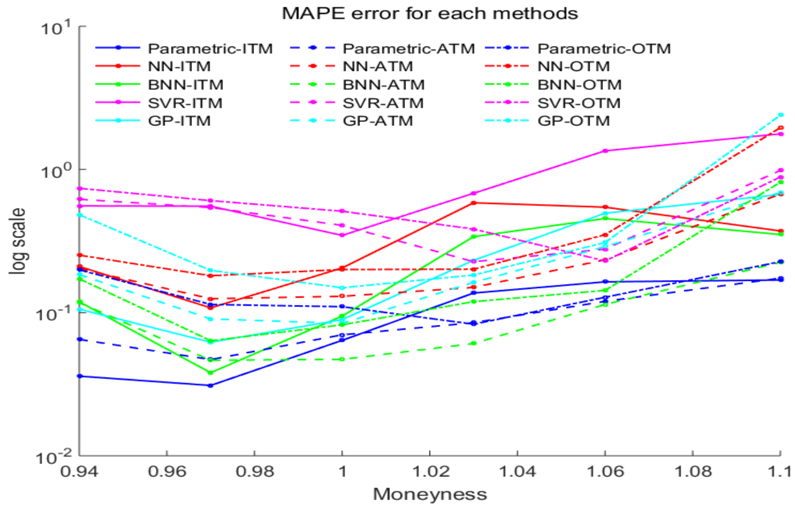

Figure 6: Total MAPE 1-day prediction error with respect to moneyness and expiration date.

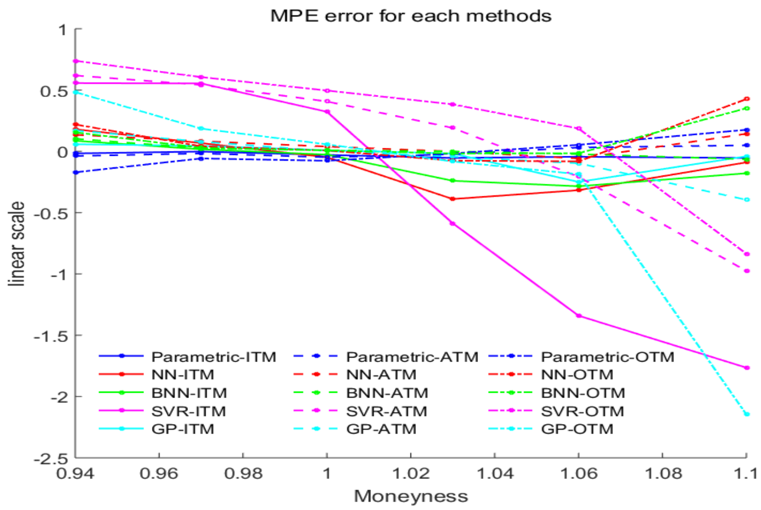

Figure 7: Total MPE 1-day prediction error with respect to moneyness and expiration date.

cess 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.

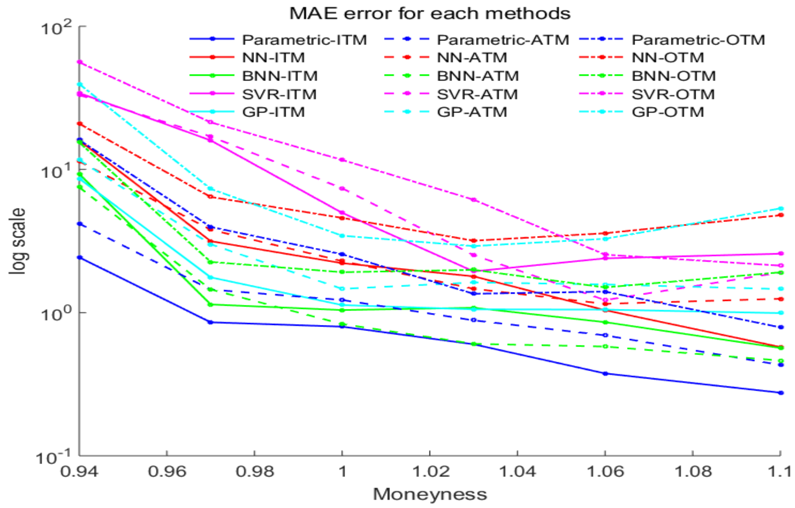

Figure 8: Total MAE 1-day prediction error with respect to moneyness and expiration date.

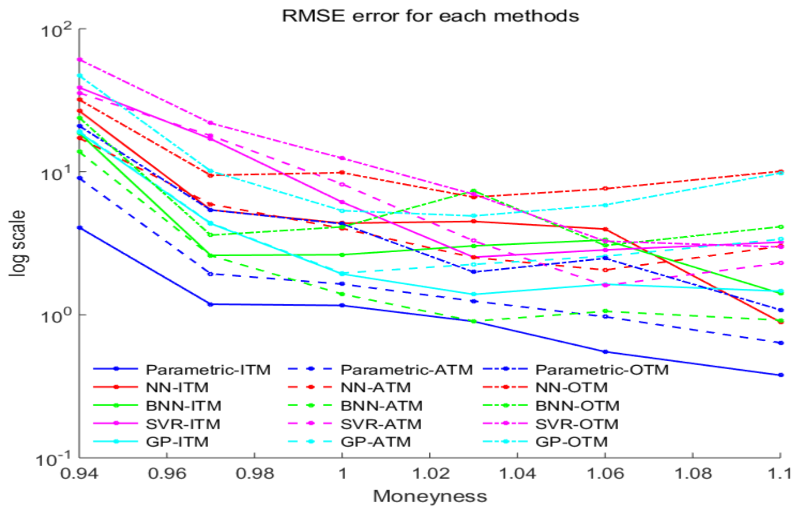

Figure 9: Total RMSE 1-day prediction error with respect to moneyness and expiration date.

4.3. Robustness 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

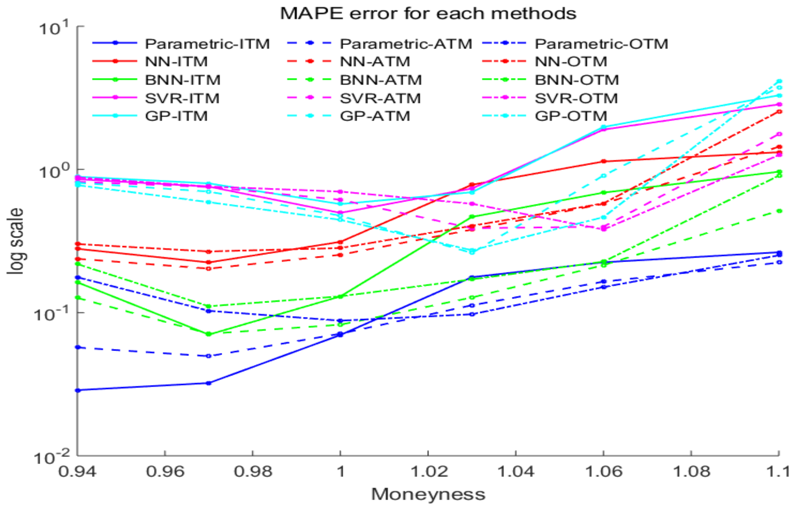

Figure 10: Total MAPE 7-day prediction error with respect to moneyness and expiration date.

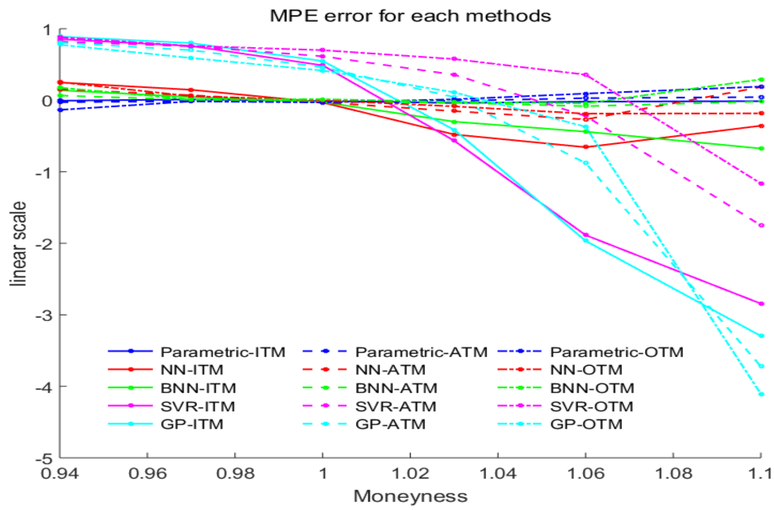

Figure 11: Total MPE 7-day prediction error with respect to moneyness and expiration date.

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 dimen-

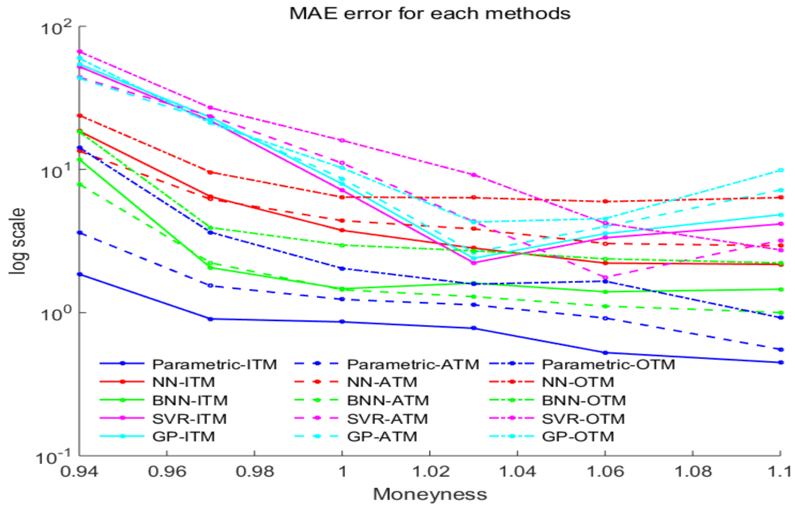

Figure 12: Total MAE 7-day prediction error with respect to moneyness and expiration date.

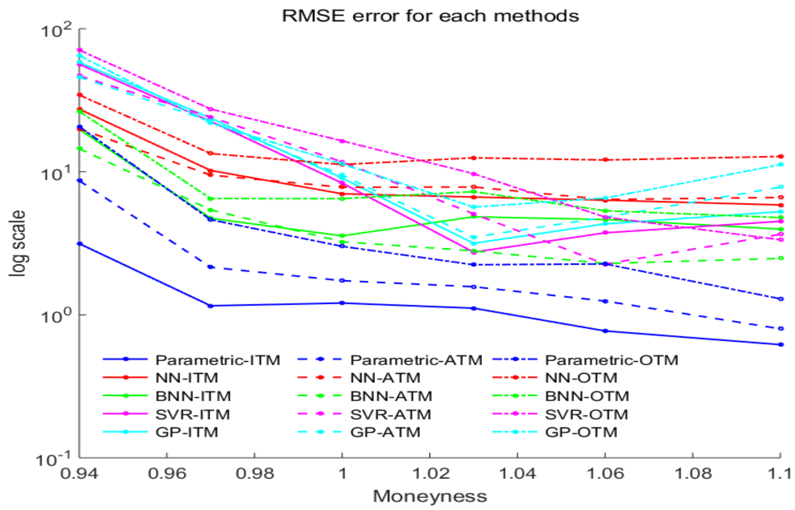

Figure 13: Total RMSE 7-day prediction error with respect to moneyness and expiration date.

sional, A multidimensional scaling method (MDS) [37] 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.

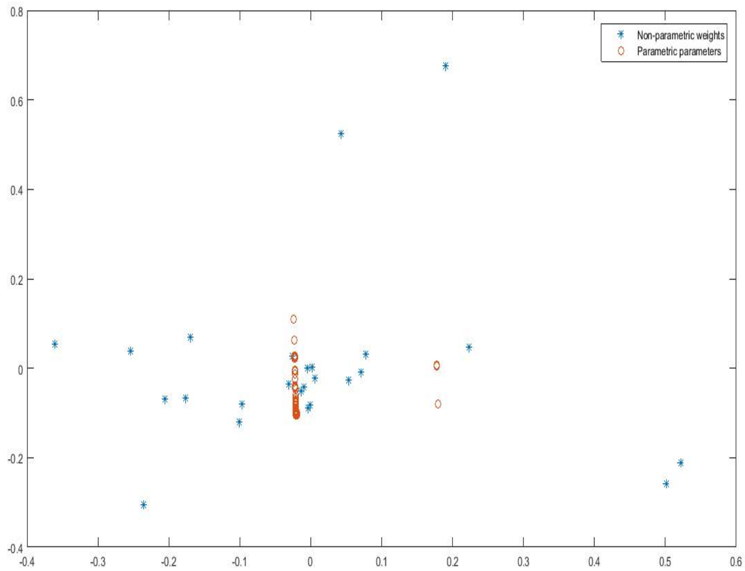

Figure 14: Two dimensional MDS visualization of each parameters. Red 'o': corresponding 1-day parametric parameters. Blue '*': corresponding 1-day nonparametric parameters.

Parameter set for representing the model is estimated through the estimation step using a daily options data and updated according the update of the daily data for calibration. We assume that it is reasonable to have no rapid changes in the model during the day interval. Therefore, we define the robustness of the estimated parameters as the absence of a rapid change between the parameter estimates of each step and evaluate each model from the robustness. 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 14 shows a typical plot for the 2D MDS visualization of the econometric jump model and the machine learning model. Each point in Figure 14 is the estimated daily parameter calibrated from the daily updated option prices. The two types of models present a different trend. Daily updated Kou model parameters, represented by red ”o”s, are mostly confined to a small range of regions. Daily updated 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.

4.4. Domain Adaptation Performance

A domain is defined to apply domain adaptation. A domain D contains two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = x_1, ..., x_n \in \mathcal{X}$. As an example, x_i is the $i-th$ input variables, such as an expiration date, a moneyness, and a price of underlying, $\mathcal{X}$ is the space of all terms, and X is the feature collection of training data for American option pricing problems. In this sense, different types of financial derivatives have different domains: each domain reflects different demands for the derivatives and considers various investor's interests arising from the different payoffs and information they acquire. The needs for domain adaptation in financial option markets naturally arises when different types of options have the same underlying asset. Hence, it is logical to adopt a model calibrated from one type of options to predict another type of options. We compared the domain adaptation performance of each model by predicting American put option prices of the following day using estimated model from current European put option prices.

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[38, 39]. 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[40, 41].

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 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.

Panel A : 1-day prediction errors by training with European put option					
Model	training day	MAPE	MPE	MAE	RMSE
Kou		0.1517	-0.0244	1.0418	1.8085
CGMY		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	10.5747

Table 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 [38, 39] 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 [38, 39]); 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 [40, 41]. For financial derivative pricing purposes, domain adaptation is possible without the introduction of additional technologies under the explicit underlying process.

5. Conclusion

In this study, 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 from 2012. 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 future research.

References

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

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

[3] M. Malliaris, L. Salchenberger, Using neural networks to forecast the S&P 100 implied volatility, Neurocomputing 10 (2) (1996) 183–195.

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

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

[6] M. Goykhman, A. Teimouri, Machine learning in sentiment reconstruction of the simulated stock market, Physica A: Statistical Mechanics and its Applications 492 (2018) 1729–1740.

[7] L. Milačić, S. Jović, T. Vujović, J. Miljković, Application of artificial neural network with extreme learning machine for economic growth estimation, Physica A: Statistical Mechanics and its Applications 465 (2017) 285–288.

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

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

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

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

[12] H. Park, J. Lee, Forecasting nonnegative option price distributions using Bayesian kernel methods, Expert Systems with Applications 39 (18) (2012) 13243–13252.

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

[14] P. Carr, H. Geman, D. B. Madan, M. Yor, Stochastic volatility for Lévy processes, Mathematical Finance 13 (3) (2003) 345–382.

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

[16] D. B. Madan, E. Seneta, The variance gamma (VG) model for share market returns, Journal of business (1990) 511–524.

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

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

[19] D. Nualart, W. Schoutens, et al., Backward stochastic differential equations and Feynman-Kac formula for Lévy processes, with applications in finance, Bernoulli 7 (5) (2001) 761–776.

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

[21] N. Kim, Y. Lee, Estimation and prediction under local volatility jump–diffusion model, Physica A: Statistical Mechanics and its Applications 491 (2018) 729–740.

[22] H. Niu, J. Wang, Financial time series prediction by a random data-time effective RBF neural network, Soft Computing 18 (3) (2014) 497–508.

[23] A. Gregoriou, J. Healy, C. Ioannidis, Hedging under the influence of transaction costs: An empirical investigation on FTSE 100 index options, Journal of Futures Markets 27 (5) (2007) 471–494.

[24] W. Kristjanpoller, A. Fadic, M. C. Minutolo, Volatility forecast using hybrid neural network models, Expert Systems with Applications 41 (5) (2014) 2437–2442.

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

[26] C. E. Rasmussen, C. K. Williams, Gaussian processes for machine learning, vol. 1, MIT press Cambridge, 2006.

[27] H. Park, N. Kim, J. Lee, Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over KOSPI 200 Index options, Expert Systems with Applications 41 (11) (2014) 5227–5237.

[28] X.-Y. Qian, S. Gao, Financial Series Prediction: Comparison Between Precision of Time Series Models and Machine Learning Methods, arXiv preprint arXiv:1706.00948 .

[29] S. P. Das, S. Padhy, A new hybrid parametric and machine learning model with homogeneity hint for European-style index option pricing, Neural Computing and Applications 28 (12) (2017) 4061–4077.

[30] L. Fatone, F. Mariani, M. C. Recchioni, F. Zirilli, An explicitly solvable multiscale stochastic volatility model: Option pricing and calibration problems, Journal of Futures Markets 29 (9) (2009) 862–893.

[31] P. Carr, D. Madan, Option valuation using the fast Fourier transform, Journal of computational finance 2 (4) (1999) 61–73.

[32] C. Rama, T. Peter, Financial Modelling with jump processes, 2003.

[33] P. Tankov, Financial modelling with jump processes, vol. 2, CRC press, 2003.

[34] R. Cont, E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, SIAM Journal on Numerical Analysis 43 (4) (2005) 1596–1626.

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

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

[37] J. M. Lattin, J. D. Carroll, P. E. Green, Analyzing multivariate data, Thomson Brooks/Cole Pacific Grove, CA, 2003.

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

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

[40] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2066–2073, 2012.

[41] N. Patricia, B. Caputo, Learning to learn, from transfer learning to domain adaptation: A unifying perspective, in: Proceedings of the Computer Vision and Pattern Recognition, EPFL-CONF-198443, 2014.