International Journal of Smart Computing and Artificial Intelligence
International Institute of Applied Informatics
2021, Vol. 5, No. 1, 1 – 18

Predictive Uncertainty in Neural Network-Based Financial Market Forecasting

Iwao Maeda [*], Hiroyasu Matsushima [*],
Hiroki Sakaji [*], Kiyoshi Izumi [*],
David deGraw [†], Atsuo Kato [‡], Michiharu Kitano [‡]

Abstract

In financial market forecasting, various methods based on statistical analysis and neural networks have been proposed. Accurate forecasting of future market states can be helpful in decision-making related to investment behavior; however, existing forecasting methods have considerable deficiencies due to the nature of financial markets and their complexity, influenceability, and nonstationarity. Forecasting of complex systems, such as financial markets, should be performed considering predictive uncertainty, and decision-making needs to be adjusted accordingly. In the present study, we introduce the concept of uncertainty to neural network-based financial market forecasting. A sparse variational dropout Bayesian neural network (SVDBNNs) is used for stochastic prediction, and on this basis, the corresponding decision-making process is proposed. The proposed method is validated by conducting investment simulation on the historical orderbook data from the Tokyo Stock Exchange and is confirmed to enable more efficient and safe investments compared with the considered alternative approaches.

Keywords: Financial data mining, Financial market forecasting, Uncertainty consideration, Neural networks

1 Introduction

Financial market forecasting has a long history of related research, and various methods to predict future market states have been proposed. Conventionally, the methods based on the knowledge on financial engineering have been investigated [1] [2]. In recent years, more advanced approaches, such as methods based on neural networks, have been actively studied [3] [4]. High generalization ability of neural networks [5] can improve the prediction accuracy.

In spite of large number of related studies, achieving perfectly accurate prediction of future market states is still unrealistic. Financial markets are known as complex systems [6]

[*] The University of Tokyo, Tokyo, Japan
[†] Daiwa Securities Co. Ltd., Tokyo, Japan
[‡] Daiwa Institute of Research Ltd., Tokyo, Japan

[7], which do not imply deterministic relationships between the current and future states of markets. Therefore, prediction has to be performed considering uncertainty [8].

In the financial market field, future forecasting can be applied to facilitate decision-making processes related to investment behavior. Market follower and contrarian strategies are one of the simplest forecasting-based decision-making-approaches. However, in the case when financial market forecasting includes uncertainty, decision-makings based on uncertain forecasts may cause risk of losses.

Therefore, while developing a neural network-based financial market forecasting method, it is necessary to account for uncertainty. However, previous neural network-based methods aimed to perform financial market forecasting have not considered predictive uncertainty. To address this issue, in the present study, we introduce an approach based on considering uncertainty in the neural network-based financial market forecasting and decision-making processes. Predictive uncertainty is incorporated by using Bayesian neural networks (BNNs) [9] [10], which allows accounting for the posterior distribution of output variables. To implement Bayesian inference in neural networks, we apply sparse variational dropout [11] [12]. Moreover, we design a decision-making process based on the predicted distribution provided by BNNs.

The proposed method was validated by conducting investment simulation on the historical stock order data obtained from the Tokyo Stock Exchange. As a result, we confirmed that the proposed method allowed performing more efficient and safe decision makings compared with ordinary neural network based investment strategies.

The main contributions of the present study are as follows:

1. We proposed an investment decision-making process based on a Bayesian neural network to consider predictive uncertainty of financial market forecasting.

2. The proposed decision-making process was evaluated by conducting investment simulation on the historical stock order data from the Tokyo Stock Exchange. The proposed process was confirmed to provide efficient decisions.

The rest of the paper is organized as follows. In Section 2, previous research works related to the considered topic are summarized. In Section 3, we described the BNNs implemented to consider predictive uncertainty. In Section 4, an investment decision-making process using BNNs is proposed. In Section 5, the input features and neural network configuration are described. In Section 6, experimental details and results are discussed, and Section 7 outlines the final conclusions.

2 Related Work

2.1 Financial Market Forecasting

Financial market forecasting has been investigated for a long time, and there are numerous research works [13] [14]. Specially, forecasting of financial market prices and volatility is considered to be one of the most widely researched topic [1] [15]. In recent years, employing neural networks in financial market forecasting [4] [16] has been one of mainstream approaches, and a number of related topics have been investigated, such as classification-based financial markets prediction [17], forecasting stock prices based on the limit order-book data using convolutional neural networks [18], financial market prediction using long

short-term memory [19], and developing a financial trading model based on stock bar chart image time series data [20].

2.2 Investment Decision Making

Decision-making processes related to investment behavior have been investigated by conducting artificial market simulation and applying financial engineering [21]. Conventionally, dynamic model-based approaches apply simple rules and formulas to describe the trader behavior patterns, such as the sniping [22], the Zero intelligence [23], and the risk-based bidding strategies [24]. In recent years, reinforcement learning methods [25] [26], specially deep reinforcement learning (DRL) methods [27] [28] have been widely applied to learning investment strategies in various studies [29] [30], such as DRL for financial portfolio management [31], market making via reinforcement learning [32], and DRL for price trailing [33].

2.3 Predictive Uncertainty

Considering predictive uncertainty is essential in constructing forecasting models [34] [35]. Conventionally, probabilistic models, such as Gaussian process (GP) [36] have been developed to address this problem. In recent years, as a result of rapid advance in the field of neural networks, studies focused on applying them to the task of forecasting have been conducted, including such applications as uncertainty estimation via prior networks [37], evidential deep learning [38], and evaluating predictive uncertainty of neural networks under the dataset shift [39].

3 Bayesian Neural Networks (BNNs)

In general, a regression model outputs one value for one certain input variable. In contrast, a model, which can consider predictive uncertainty has to output more information, including posterior distribution of the output variable [34]. In the present study, BNNs are employed to account for predictive uncertainty.

Conceptually, BNNs [9] [10] perform Bayesian inference in neural networks by approximating the following posterior distribution.

$$p(y|D,x,\theta) = \int p(y|x,w)p(w|D,\theta) \tag{1}$$

where x and y are input and output variables; D is the training data; θ denotes the set of parameters of the neural network; and w corresponds to the set of weight values conditioned by D and θ. By approximating the posterior distribution of the output variable, we can incorporate predictive uncertainty. Narrow tail distribution is predicted in the case when the predictive uncertainty is small, and fat tail distribution is obtained when the predictive uncertainty is large.

There are several methods developed for training BNNs, such as Laplace approximation [40], variational inference [41] [10], and automatic differentiation variational inference (ADVI) [42]. In the present study, the method based on sparse variational dropout is used. Variational dropout is employed to interpret Gaussian dropout [43] as a Bayesian approximation. By applying Gaussian dropout also in the inference phase, sampling from

the approximated posterior distribution of the output variable can be realized [44]. Optimization of parameters in BNNs can be performed using the stochastic gradient variational Bayes (SGVB) method [11] [45]. By using SGVB, an unbiased Monte Carlo estimator of the expected log likelihood can be obtained. Moreover, the local reparameterization trick proposed in [11], as well as reparameterization of posterior parameters and approximation of Kullback–Leibler (KL) divergence proposed in [12] can be used to improve variational dropout aiming to perform learning with a small variance. For the purpose of the present study, we employ sparse variational dropout Bayesian neural networks (SVDBNNs) proposed in [12]. SVDBNNs are used to parameterize mean μ_i and standard deviation σ_i of each weight w_i in networks instead of the dropout rate α_i aiming to reduce the variance of a gradient. Each weight value is sampled according to the following equation:

$$w_i = \mu_i + \sigma_i \varepsilon_i \tag{2}$$

where $\varepsilon_i \sim N(0,1)$. α can be calculated as $\alpha_i = \sigma_i^2/\mu_i^2$. The Kullback-Leibler (KL) divergence of the parameters can be approximated as per the following equation:

$$-D_{KL} \approx k_1 \sigma(k_2 + k_3 \log \alpha_i) - 0.5 \log(1 + \alpha_i^{-1}) + C \tag{3}$$

where $k_1 = 0.63576$; $k_2 = 1.87320$; $k_3 = 1.48695$; and σ denotes the likelihood function of the standard normal distribution. The overall loss function to minimize can be written as follows:

$$-L_D(\theta) + D_{KL} \tag{4}$$

where $-L_D(\theta)$ is the negative log likelihood for the training data and is equal to the cross entropy function.

SVDBNNs can be used for prediction to obtain the outputs stochastically. Prediction in BNNs is performed by sampling parameters of the network from their posterior distributions (Equation 2), and the posterior distribution of the output variable cannot be identified analytically. Therefore, posterior mean and standard deviation are estimated by performing multiple samplings.

4 Decision-Making Using BNNs

Predicted future prices can be used for the purpose of decision-making related to investment behavior. Applying the prediction model that can consider predictive uncertainty, the decision-making process can be improved. In the present study, we propose a baseline process using only predicted prices, and develop an improved method that incorporates both predicted prices and uncertainty.

We consider that a prediction model $f(\cdot)$ outputs the score (or probability) values of changing in the mid price $y = [y_{\text{up}}, y_{\text{stay}}, y_{\text{down}}]$, where ($y_{\text{up}}$, y_{stay}, and y_{down}) are the scores of a price increase, leveling off, and fall, respectively, and $y_{\text{up}} + y_{\text{stay}} + y_{\text{down}} = 1$. We note that price fluctuations below a predefined threshold are as leveling off (the threshold is set according to orderbook states). We denote this prediction as $y = f(X)$, where X correspond to a set of market states.

4.1 Score-Based Process

As a baseline, we propose a score-based process, which takes the predicted scores as input. In the score-based process, investment behavior is described as follows:

1. Buy ($y_{\text{up}} > \max(t, y_{\text{stay}}, y_{\text{down}})$)
2. Sell ($y_{\text{down}} > \max(t, y_{\text{up}}, y_{\text{stay}})$)
3. Stay (otherwise)

where t is the score threshold. When the investment behavior corresponds to buying or selling, a transaction is executed. There are various ways to execute a transaction, such as limit order (LMT) and market order (MKT) transactions. In the present study, we consider mid price transaction to eliminate transaction costs, which allows ignoring differences in the transaction frequency. In the case of executing mid price transaction with price p_{mid} and volume v, the change in trader cash Δc is written as $-p_{\text{mid}}v$. We note that v is positive when the behavior is buy, and negative when the behavior is sell. The investment volume v is calculated as follows:

$$v = \begin{cases} \frac{(c+p_{\text{mid}}I)r}{p_{\text{mid}}} & (\text{Buy}) \\ -\frac{(c+p_{\text{mid}}I)r}{p_{\text{mid}}} & (\text{Sell}) \end{cases} \tag{5}$$

where I and r are the inventory value and investment ratio, respectively. $c + p_{\text{mid}}I$ represents the current capital value of a trader. In addition, the limit of absolute inventory I_{lim} is considered. The inventory I must be within $-I_{\text{lim}}$ and I_{lim}. Then, I_{lim} is calculated as follows:

$$I_{\text{lim}} = \frac{(c + p_{\text{mid}}I)l}{p_{\text{mid}}} \tag{6}$$

where l is the leverage value. Then, the investment volume v can be calculated as follows:

$$v = \begin{cases} \min(\frac{(c+p_{\text{mid}}I)r}{p_{\text{mid}}}, \max(I_{\text{lim}} - I, 0)) & (\text{Buy}) \\ -\min(\frac{(c+p_{\text{mid}}I)r}{p_{\text{mid}}}, \max(I_{\text{lim}} + I, 0)) & (\text{Sell}) \end{cases} \tag{7}$$

The algorithm of the score-based process is described in Algorithm 1. In case of a standard neural network, the deterministically predicted scores are used. In turn, in case of a Bayesian neural network, the mean values of sampled scores are used as the predicted scores.

4.2 Std-Based Process

By accounting for predicted uncertainty, decision-making processes can be improved. Here, we propose a process based on standard deviation (std), which uses modified scores calculated as follows:

$$y' = y - k\sigma_y \tag{8}$$

where k is a coefficient, and σ_y is standard deviation of y. σ_y can be substituted by sample standard deviation in cases when standard deviation cannot be obtained analytically (as

Algorithm 1 Decision-making and investment behavior in the score-based process

Require: prediction NN (BNN) $f(\cdot)$
Parameter: score threshold t, investment ratio r, and leverage l
Get current cash c and inventory I
Get market states X and mid price p_{mid}
Predict scores using NN (BNN) $[y_{\mathrm{up}}, y_{\mathrm{stay}}, y_{\mathrm{down}}] = f(X)$
Get inventory limit $I_{\mathrm{lim}} = \frac{(c+p_{\mathrm{mid}}I)l}{p_{\mathrm{mid}}}$
if $y_{\mathrm{up}} > \max(t, y_{\mathrm{stay}}, y_{\mathrm{down}})$ **then**
  Determine buy volume $v = \min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} - I, 0))$
else if $y_{\mathrm{down}} > \max(t, y_{\mathrm{up}}, y_{\mathrm{stay}})$ **then**
  Determine sell volume $v = -\min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} + I, 0))$
else
  Do nothing $v = 0$
end if
Trade on the current mid price and the determined volume
Update cash $c \leftarrow c - p_{\mathrm{mid}}v$
Update inventory $I \leftarrow I + v$

in BNN). Equation 8 is based on the lower confidence bound [46] [47]. The lower or upper confidence bound has been proposed as an acquisition function for Bayesian optimization [48] [49], and is known to enable efficient parameter search. The coefficient k represents the trade-off between utilization and search in optimization. In the present study, a larger k corresponds to safer investment decisions.

After calculating y', the rest of the procedure is the same as in the score-based process. The algorithm of the std-based process is shown in Algorithm 2.

5 Neural Network Configuration

In this section, we describe feature engineering, and the network architecture of the proposed neural network.

5.1 Feature Engineering

As the feature input into neural networks, two features; price series and orderbook features, are used.

The price series is composed of the data corresponding to ten market prices (the last or current trade price) registered at certain time step intervals. In this study, the interval is set to ten tick.

The orderbook features are arranged in a vector of the latest orderbook and are used to summarize order volumes of the upper and lower prices centered at the mid price. To distinguish between the buy and sell orders, buy order volumes are recorded as negative values.

The positional information of the orderbook features is considered important for the price trend prediction, but traditional multivariate analysis methods cannot extract positional information. As described in the next section, we applied convolutional neural networks (CNNs) for valid predictions.

Algorithm 2 Decision making and investment behavior of the std-based process

Require: prediction BNN $f(\cdot)$
Parameter: coefficient k, score threshold t, investment ratio r and leverage l
Get current cash c and inventory I
Get market states X and mid price p_{mid}
for $i = 1, \cdots, N$ **do**
  Sample scores using BNN $[y_{i,\mathrm{up}}, y_{i,\mathrm{stay}}, y_{i,\mathrm{down}}] \sim f(X)$
end for
Get modified sores $y'_{\mathrm{up}} = E[y_{i,\mathrm{up}}] - k\sqrt{V[y_{i,\mathrm{up}}]}, y'_{\mathrm{stay}} = E[y_{i,\mathrm{stay}}] - k\sqrt{V[y_{i,\mathrm{stay}}]}$,
$y'_{\mathrm{down}} = E[y_{i,\mathrm{down}}] - k\sqrt{V[y_{i,\mathrm{down}}]}$
Get inventory limit $I_{\mathrm{lim}} = \frac{(c+p_{\mathrm{mid}}I)l}{p_{\mathrm{mid}}}$
if $y'_{\mathrm{up}} > \max(t, y'_{\mathrm{stay}}, y'_{\mathrm{down}})$ **then**
  Determine buy volume $v = \min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} - I, 0))$
else if $y'_{\mathrm{down}} > \max(t, y'_{\mathrm{up}}, y'_{\mathrm{stay}})$ **then**
  Determine sell volume $v = -\min(\frac{(c+p_{\mathrm{mid}}I)r}{p_{\mathrm{mid}}}, \max(I_{\mathrm{lim}} + I, 0))$
else
  Do nothing $v = 0$
end if
Trade on the current mid price and the determined volume
Update cash $c \leftarrow c - p_{\mathrm{mid}}v$
Update inventory $I \leftarrow I + v$

5.2 Network Architecture

The proposed network architecture is represented in Figure 1. The baseline and BNNs have the same layer configuration. In these networks, two market state features, price series, and orderbook features (Section 5.1), are extracted and merged. To extract price series features, a long short-term memory (LSTM) [50] layer is employed according to the previous studies [19] [51]. Convolutional (Conv) and maximu pooling (MaxPool) layers [52] are used to extract the orderbook features that correspond to the positional information [18] [53]. Merged features are transformed by using fully-connected (Dense) layers, and the price change y is the output from the last layer.

Convolutional and fully-connected layers can be converted to the sparse variational dropout (SVD) form [12]. Therefore, in the proposed BNN, SVD convolutional, and fully-connected layers are used instead of ordinary ones.

6 Experiments

To validate the developed methods, several experiments were conducted. For this purpose, the proposed models were trained using the historical stock order data obtained from the Tokyo Stock Exchange. The model performance was validated by checking prediction accuracies and investment performance.

In these experiments, baseline and Bayesian NN models were considered. In addition, the models with L2 regularization (L2Regu) [54] [55] were also trained using the same procedure. L2 regularization is a technique commonly used to prevent overfitting [56] [57] and to improve the prediction accuracy. In the present study, L2 regularization was applied

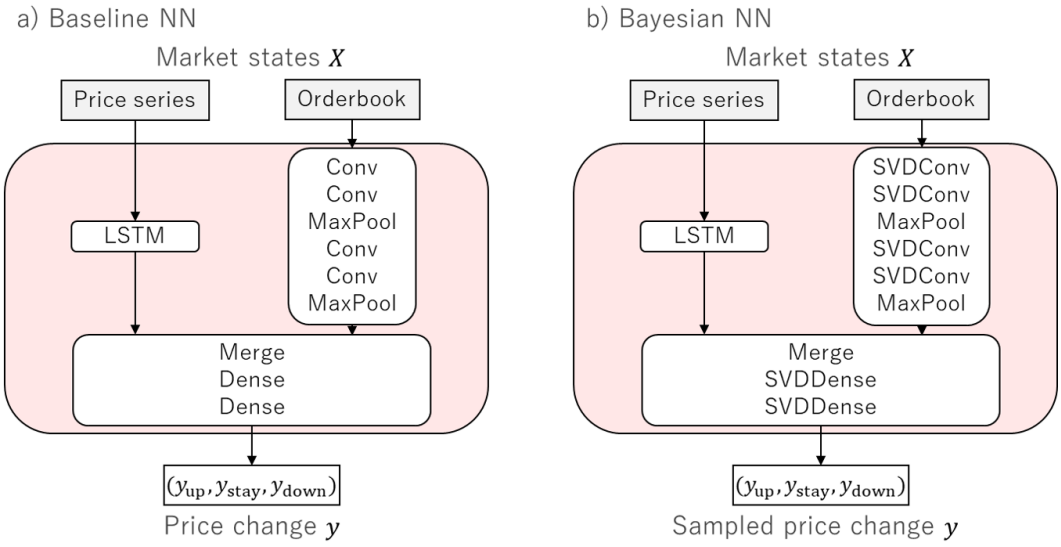

Figure 1: Overview of the proposed networks. Two market state features, price series, and orderbook features are extracted and merged. The LSTM layer is used to extract the price series features, and the Conv and MaxPool layers are employed to extract orderbook features. Merged features are transformed by using Dense layers, and the price change is obtained as the final output. In the proposed BNN, SVD Conv, and Dense layers are employed instead of ordinary ones to convert the network into the Bayesian form.

to the parameters in the LSTM, convolutional, and fully-connected layers.

6.1 Dataset

To conduct the experiments, we used the FLEX_FULL historical full orderbook data obtained from the Tokyo Stock Exchange [1]. The FLEX_FULL dataset contains tens of millions of stock order data items per day including high-frequency trading orders [58].

In this experiment, the data for symbol 9022 (Central Japan Railway Company) collected in the period between January 1, 2019 and June 30, 2019 were used for training, and data collected between July 1, 2019 and August 31, 2019 was used for validation. Training samples were extracted picking every ten available samples to reduce correlation between samples.

6.2 Prediction Accuracy

Prediction accuracy values estimated on the validation data are represented in Table 1. Predicted price change probabilities are validated in terms if the area under the receiver operating characteristics (AUROC) [59], false-positive rate 95% (FPR95), area under precision-recall curve (AUPR) [60] [61], and expected calibration error (ECE) [62]. ECE is defined as the average difference between the predicted score and accuracy and is used to measure the validity of predicted scores (equal to predictive uncertainty).

As shown in Table 1, all proposed methods achieved the approximately similar values of AUROC, FPR95, and AUPR. This result is in line with preliminary expectations, as

[1] https://www.jpx.co.jp/english/markets/paid-info-equities/realtime/index.html

Table 1: Prediction accuracy estimated on the validation data. Predicted order probabilities are validated in terms of AUROC, FPR95, AUPR, and ECE.

Model	AUROC ↑	FPR95 ↓	AUPR ↑	ECE ↓
Baseline	**0.5627**	0.9322	0.3767	0.1087
Baseline + L2Regu	0.5599	0.9336	0.3727	0.1145
SVDBNN	0.5504	0.9167	0.3725	**0.0068**
SVDBNN + L2Regu	0.5603	**0.9081**	**0.3819**	0.0089

no modification has been made to improve the prediction accuracy. However, the proposed SVDBNN and SVDBNN + L2Regu have achieved excellent ECE values and have predicted uncertainty much more accurately compared with the than baseline NN and NN + L2Regu.

6.3 Investment Performance

Investment performance estimates are provided in Table 2. Investment simulation was performed for each day in the validation dataset, and the simulation procedure was implemented according to Algorithm 1 and Algorithm 2. Traders were given possibility to invest on every 10 order, and act according to the predictions of the trained models. Here, we set investment ratio $r = 0.1$, and leverage $l = 1$. In addition, score threshold t was selected in the range of $\{1/3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ ($1/3$ is the minimum possible value of $\max(y_{\text{up}}, y_{\text{stay}}, y_{\text{down}})$) in the case of the baseline NN, and in the range of $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ in the case of the Bayesian NN. In the std-based process described in Section 4.2, the coefficient value k was also changed in the range of $\{0.5, 1, 2\}$. The initial cash and inventory values were set to 1.0×10^8 and 0.

Investment performance estimates were evaluated in terms of the total return rate T_r, Sharpe ratio S_p [63] [64], and maximum drawdown MDD [65] [66]. The total return rate T_r is calculated as follows:

$$T_r = \frac{cap_{\text{end}} - cap_{\text{start}}}{cap_{\text{start}}} \tag{9}$$

where cap_{start} and cap_{end} are the initial and final capital values calculated by $cap = c + p_{\text{mid}} I$. Daily T_r values are multiplied by 250 to be converted to the annual rates. The Sharpe ratio S_p is calculated as follows:

$$S_p = \frac{E[R_a - R_b]}{\sqrt{V[R_a - R_b]}} \tag{10}$$

where R_a and R_b are returns of the investment and benchmark, and E and V are expected value and variance. In this study, $R_b = 0$ was assumed. Maximum drawdown is calculated using the following equation:

$$MDD = \frac{P - L}{P} \tag{11}$$

where P and L are the highest and lowest capital amounts before and after the largest capital drop.

As shown in Table 2 and Table 3, all models achieved acceptable T_r and S_p values in the case when the threshold t was small. In general, the large value of t allows preventing large losses, but limits the possibility of gaining stable profits at the same time. In addition, the

Table 2: Investment performance estimates of the score-based process using the trained baseline NN and SVDBNN models. Performance estimates of the models are validated in terms of the total return rate T_r, Sharpe ratio S_p, and maximum drawdown *MDD*. Mean and standard deviation values of daily indices are provided. The rows without any values indi-cated the cases when no simulation could be performed due to excessively large threshold values.

Model	t	$T_r \uparrow$	$S_p \uparrow$	$MDD \downarrow$
Baseline	$1/3$	0.9379 ± 1.3388	0.0138 ± 0.0197	0.0016 ± 0.0004
(Score based)	0.4	0.8338 ± 1.3769	0.0121 ± 0.0194	0.0016 ± 0.0004
	0.5	0.5159 ± 1.1397	0.0076 ± 0.0170	0.0017 ± 0.0004
	0.6	0.1867 ± 1.3677	0.0020 ± 0.0199	0.0018 ± 0.0004
	0.7	0.0655 ± 1.3935	0.0006 ± 0.0205	0.0015 ± 0.0006
	0.8	-0.0573 ± 1.1387	-0.0033 ± 0.0212	0.0012 ± 0.0006
Baseline	$1/3$	1.2575 ± 1.6847	0.0174 ± 0.0206	0.0014 ± 0.0002
+ L2Regu	0.4	1.1952 ± 1.7257	0.0162 ± 0.0210	0.0014 ± 0.0003
(Score based)	0.5	0.8636 ± 1.7033	0.0102 ± 0.0215	0.0016 ± 0.0003
	0.6	0.4135 ± 1.4812	0.0031 ± 0.0199	0.0017 ± 0.0005
	0.7	0.3654 ± 1.5046	0.0027 ± 0.0232	0.0015 ± 0.0006
	0.8	0.2087 ± 1.1344	0.0048 ± 0.0236	0.0011 ± 0.0006
SVDBNN	$1/3$	1.1951 ± 1.5409	0.0169 ± 0.0207	0.0013 ± 0.0003
(Score based)	0.4	0.6977 ± 1.6765	0.0080 ± 0.0228	0.0015 ± 0.0003
	0.5	-0.0024 ± 0.8229	0.0017 ± 0.0230	0.0009 ± 0.0010
	0.6	-0.1827 ± 0.7854	-0.0013 ± 0.0212	0.0007 ± 0.0014
	0.7	-0.0259 ± 0.1088	-0.0117 ± 0.0138	0.0001 ± 0.0003
	0.8	–	–	–
SVDBNN	$1/3$	1.3602 ± 1.3615	0.0208 ± 0.0197	0.0013 ± 0.0003
+ L2Regu	0.4	0.6145 ± 1.2419	0.0085 ± 0.0202	0.0015 ± 0.0003
(Score based)	0.5	-0.0487 ± 1.2490	-0.0006 ± 0.0215	0.0014 ± 0.0006
	0.6	-0.1248 ± 0.7760	-0.0018 ± 0.0261	0.0006 ± 0.0005
	0.7	0.0018 ± 0.0117	0.0100 ± 0.0000	0.0000 ± 0.0000
	0.8	–	–	–

Table 3: Investment performance estimates of the std-based process using the trained SVDBNN models. Performance estimates of the models are validated in terms of the total return rate T_r, Sharpe ratio S_p, and maximum drawdown *MDD*. Mean and standard deviation values of daily indices are provided. The rows without any values indicated the cases when no simulation could be performed due to excessively large threshold values.

Model	k	t	$T_r \uparrow$	$S_p \uparrow$	$MDD \downarrow$
SVDBNN	0.5	0	1.2450 ± 1.4619	0.0177 ± 0.0191	0.0013 ± 0.0003
(Std based)		0.1	1.3119 ± 1.4066	0.0192 ± 0.0190	0.0013 ± 0.0002
		0.2	1.2634 ± 1.4551	0.0180 ± 0.0194	0.0013 ± 0.0002
		0.3	1.2748 ± 1.6564	0.0175 ± 0.0208	0.0013 ± 0.0002
		0.4	0.5462 ± 1.7732	0.0052 ± 0.0222	0.0018 ± 0.0005
		0.5	-0.2973 ± 0.9135	-0.0053 ± 0.0194	0.0009 ± 0.0015
SVDBNN	1	0	1.2243 ± 1.5536	0.0169 ± 0.0199	0.0013 ± 0.0002
(Std based)		0.1	1.2976 ± 1.5561	0.0182 ± 0.0206	0.0013 ± 0.0002
		0.2	1.2374 ± 1.6341	0.0170 ± 0.0208	0.0013 ± 0.0002
		0.3	1.2111 ± 1.4363	0.0170 ± 0.0192	0.0014 ± 0.0003
		0.4	-0.4022 ± 1.1832	-0.0058 ± 0.0190	0.0013 ± 0.0015
		0.5	-0.2971 ± 0.8451	-0.0061 ± 0.0196	0.0008 ± 0.0015
SVDBNN	2	0	1.0663 ± 1.6193	0.0138 ± 0.0208	0.0015 ± 0.0006
(Std based)		0.1	1.2235 ± 1.6485	0.0163 ± 0.0212	0.0015 ± 0.0005
		0.2	1.0241 ± 1.5412	0.0136 ± 0.0206	0.0014 ± 0.0003
		0.3	0.7311 ± 1.5499	0.0091 ± 0.0212	0.0015 ± 0.0004
		0.4	-0.2767 ± 0.9115	-0.0052 ± 0.0190	0.0009 ± 0.0015
		0.5	-0.0097 ± 0.0796	-0.0020 ± 0.0200	0.0000 ± 0.0001
SVDBNN	0.5	0	1.4327 ± 1.2971	0.0224 ± 0.0196	0.0012 ± 0.0003
+ L2Regu		0.1	1.4289 ± 1.4009	0.0213 ± 0.0191	0.0013 ± 0.0003
(Std based)		0.2	1.3262 ± 1.4027	0.0201 ± 0.0198	0.0013 ± 0.0003
		0.3	1.4454 ± 1.3298	0.0224 ± 0.0195	0.0013 ± 0.0003
		0.4	-0.1249 ± 1.2795	-0.0020 ± 0.0205	0.0017 ± 0.0013
		0.5	-0.1191 ± 0.7834	-0.0042 ± 0.0221	0.0006 ± 0.0005
SVDBNN	1	0	1.3759 ± 1.3126	0.0212 ± 0.0175	0.0014 ± 0.0004
+ L2Regu		0.1	1.4013 ± 1.2687	0.0214 ± 0.0174	0.0013 ± 0.0003
(Std based)		0.2	1.4828 ± 1.5593	0.0223 ± 0.0209	0.0013 ± 0.0003
		0.3	1.4453 ± 1.3713	0.0216 ± 0.0182	0.0013 ± 0.0003
		0.4	-0.0479 ± 0.8866	-0.0092 ± 0.0224	0.0006 ± 0.0005
		0.5	–	–	–
SVDBNN	2	0	1.2509 ± 1.2408	0.0191 ± 0.0177	0.0013 ± 0.0003
+ L2Regu		0.1	1.1639 ± 1.3837	0.0173 ± 0.0193	0.0013 ± 0.0003
(Std based)		0.2	1.3471 ± 1.4619	0.0206 ± 0.0204	0.0013 ± 0.0003
		0.3	1.0053 ± 1.3677	0.0155 ± 0.0212	0.0015 ± 0.0004
		0.4	–	–	–
		0.5	–	–	–

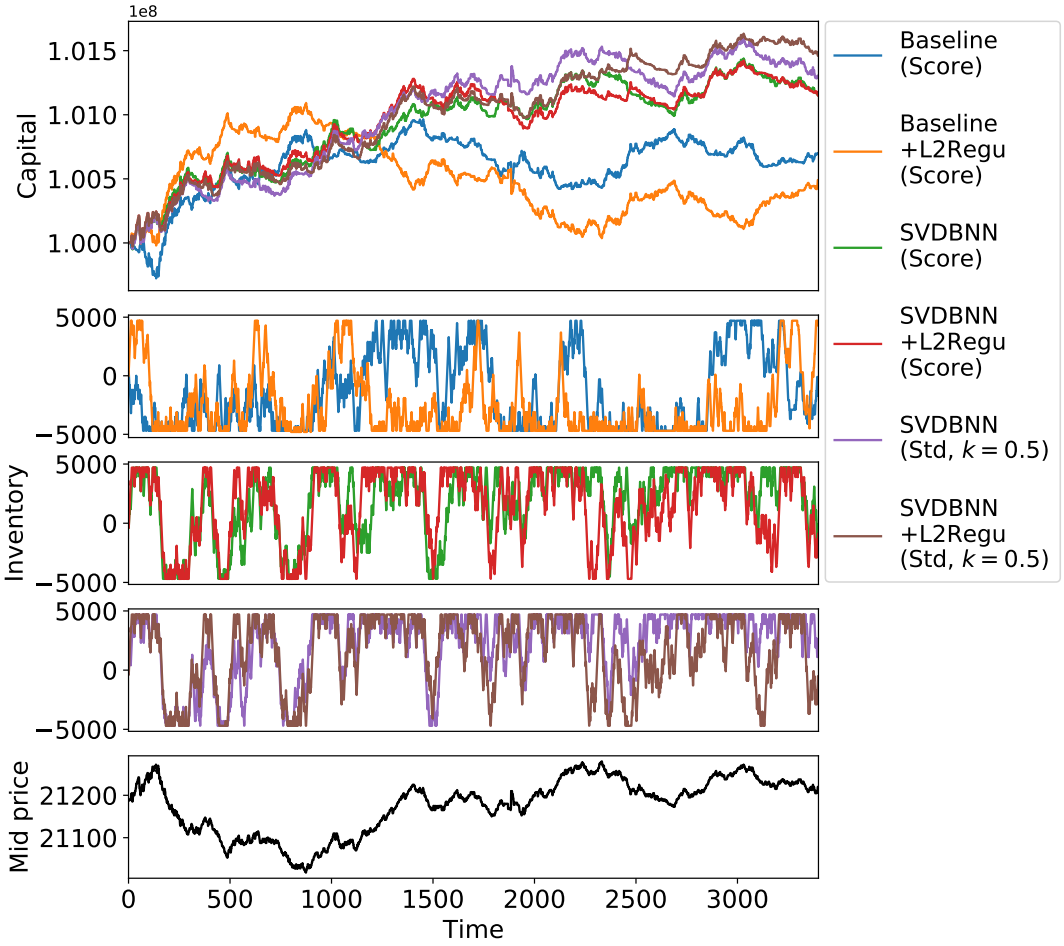

Figure 2: Example of changes in capitals and mid price (August, 8, 2019). The top plot shows changes in capitals of the proposed methods, the middle three plots show changes in inventories, and the bottom plot represents changes in the mid price. $t = 1/3$ for the score based processes; and $k = 0.5$ and $t = 0$ for the std-based process are selected as representatives. The horizontal axis represents the number of actions.

proposed std-based process using SVDBNN with l2 regularization achieved better results. By setting $k = 0.5$ and $t = 0$, effective investments with large T_r and S_p, and small *MDD* can be realized.

Example investment results are represented in Figure 2. Figure 2 depicts changes in capitals and inventories of the proposed methods, and mid price according to the number of actions. The results for the case when $t = 1/3$ are provided for the score-based processes using the baseline and SVDBNN models, and the results when $k = 0.5$ and $t = 0$ are represented for the std-based processes using SVDBNN models. As shown in Figure 2, inventories of baseline models and SVDBNN models changed in opposite directions. In this case, SVDBNN models achieved more accurate predictions compared with the baseline ones, and the investment performance estimates of the SVDBNN models were excellent. Comparing the score-based and std-based processes using SVDBNN models, it can be concluded that std-based process achieved slightly better performance.

7 Conclusions

In the present study, we aimed to demonstrate the importance of considering predictive uncertainty in neural-network based financial market prediction and corresponding decision-making. We introduced BNNs into financial market forecasting to consider predictive uncertainty, and proposed an investment decision-making process based on BNNs. Experiments were conducted using the historical stock order data, and the proposed method was confirmed to allow performing more efficient decision makings.

With regard to this advanced problem, we note that it is difficult to make decisions in the case when major changes in financial markets (like the financial crisis [67]) happen. There are greater potential risks associated with such market conditions, and decision-making processes robust with respect to these risks are required. In the future research, we will aim to update our the proposed method to consider major market changes.

References

[1] T. G. Andersen, T. Bollerslev, and S. Lange, "Forecasting financial market volatility: Sample frequency vis-a-vis forecast horizon," *Journal of empirical finance*, vol. 6, no. 5, pp. 457–477, 1999.

[2] D. W. Bunn, "Modelling prices in competitive electricity markets," 2004.

[3] E. M. Azoff, *Neural network time series forecasting of financial markets*. John Wiley & Sons, Inc., 1994.

[4] A.-S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: forecasting and trading the taiwan stock index," *Computers & Operations Research*, vol. 30, no. 6, pp. 901–923, 2003.

[5] E. Levin, N. Tishby, and S. A. Solla, "A statistical approach to learning and generalization in layered neural networks," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1568–1574, 1990.

[6] D. Sornette, *Why stock markets crash: critical events in complex financial systems*. Princeton University Press, 2017, vol. 49.

[7] W. B. Arthur, *The economy as an evolving complex system II*. CRC Press, 2018.

[8] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[9] A. Bate, M. Lindquist, I. R. Edwards, S. Olsson, R. Orre, A. Lansner, and R. M. De Freitas, "A bayesian neural network method for adverse drug reaction signal generation," *European journal of clinical pharmacology*, vol. 54, no. 4, pp. 315–321, 1998.

[10] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[11] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems*, 2015, pp. 2575–2583.

[12] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational dropout sparsifies deep neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2498–2507.

[13] E. J. Bomhoff and E. J. Bomhoff, *Financial forecasting for business and economics*. Dryden Press London, 1994.

[14] Y. S. Abu-Mostafa and A. F. Atiya, "Introduction to financial forecasting," *Applied Intelligence*, vol. 6, no. 3, pp. 205–213, 1996.

[15] S.-H. Poon, *A practical guide to forecasting financial market volatility*. John Wiley & Sons, 2005.

[16] Y. Li and W. Ma, "Applications of artificial neural networks in financial economics: a survey," in *2010 International symposium on computational intelligence and design*, vol. 1. IEEE, 2010, pp. 211–214.

[17] M. Dixon, D. Klabjan, and J. H. Bang, "Classification-based financial markets prediction using deep neural networks," *Algorithmic Finance*, vol. 6, no. 3-4, pp. 67–77, 2017.

[18] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Forecasting stock prices from the limit order book using convolutional neural networks," in *2017 IEEE 19th Conference on Business Informatics (CBI)*, vol. 1, 2017, pp. 7–12.

[19] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[20] O. B. Sezer and A. M. Ozbayoglu, "Financial trading model with stock bar chart image time series with deep convolutional neural networks," *arXiv preprint arXiv:1903.04610*, 2019.

[21] B. LeBaron, "Building the santa fe artificial stock market," *Physica A*, pp. 1–20, 2002.

[22] J. Rust, R. Palmer, and J. H. Miller, "Behaviour of trading automata in a computerized double auction market." Santa Fe Institute, 1992.

[23] D. Ladley, "Zero intelligence in economics and finance," *The Knowledge Engineering Review*, vol. 27, no. 2, pp. 273–286, 2012.

[24] P. Vytelingum, R. K. Dash, E. David, and N. R. Jennings, "A risk-based bidding strategy for continuous double auctions," in *ECAI*, vol. 16, 2004, p. 79.

[25] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[26] P. Dayan and B. W. Balleine, "Reward, motivation, and reinforcement learning," *Neuron*, vol. 36, no. 2, pp. 285–298, 2002.

[27] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[29] Y. Nevmyvaka, Y. Feng, and M. Kearns, "Reinforcement learning for optimized trade execution," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 673–680.

[30] T. L. Meng and M. Khushi, "Reinforcement learning in financial markets," *Data*, vol. 4, no. 3, p. 110, 2019.

[31] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *arXiv preprint arXiv:1706.10059*, 2017.

[32] T. Spooner, J. Fearnley, R. Savani, and A. Koukorinis, "Market making via reinforcement learning," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 434–442.

[33] K. S. Zarkias, N. Passalis, A. Tsantekidis, and A. Tefas, "Deep reinforcement learning for financial trading using price trailing," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3067–3071.

[34] B. Renard, D. Kavetski, G. Kuczera, M. Thyer, and S. W. Franks, "Understanding predictive uncertainty in hydrologic modeling: The challenge of identifying input and structural errors," *Water Resources Research*, vol. 46, no. 5, 2010.

[35] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[36] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[37] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058.

[38] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Advances in Neural Information Processing Systems*, 2018, pp. 3179–3189.

[39] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 969–13 980.

[40] W. Wright, "Bayesian approach to neural-network modeling with input uncertainty," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1261–1270, 1999.

[41] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*, 2011, pp. 2348–2356.

[42] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, "Automatic differentiation variational inference," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 430–474, 2017.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[44] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[45] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, p. II–1278–II–1286.

[46] Z. Govindarajulu, "Distribution-free confidence bounds for p (x¡ y)," *Annals of the institute of statistical mathematics*, vol. 20, no. 2, pp. 229–38, 1968.

[47] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[48] M. Pelikan, D. E. Goldberg, E. Cantú-Paz *et al.*, "Boa: The bayesian optimization algorithm," in *Proceedings of the genetic and evolutionary computation conference GECCO-99*, vol. 1, 1999, pp. 525–532.

[49] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[51] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PloS one*, vol. 12, no. 7, p. e0180944, 2017.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[53] D. Tashiro, H. Matsushima, K. Izumi, and H. Sakaji, "Encoding of high-frequency order information and prediction of short-term stock price by deep learning," *Quantitative Finance*, vol. 19, no. 9, pp. 1499–1506, 2019.

[54] A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 78.

[55] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135–1143.

[56] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.

[57] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[58] J. Brogaard, T. Hendershott, and R. Riordan, "High-frequency trading and price discovery," *The Review of Financial Studies*, vol. 27, no. 8, pp. 2267–2306, 2014.

[59] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[60] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: point estimates and confidence intervals," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 451–466.

[61] J. Keilwagen, I. Grosse, and J. Grau, "Area under precision-recall curves for weighted and unweighted data," *PloS one*, vol. 9, no. 3, p. e92209, 2014.

[62] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1321–1330.

[63] W. F. Sharpe, "The sharpe ratio," *Journal of portfolio management*, vol. 21, no. 1, pp. 49–58, 1994.

[64] O. Ledoit and M. Wolf, "Robust performance hypothesis testing with the sharpe ratio," *Journal of Empirical Finance*, vol. 15, no. 5, pp. 850–859, 2008.

[65] M. Magdon-Ismail and A. F. Atiya, "Maximum drawdown," *Risk Magazine*, vol. 17, no. 10, pp. 99–102, 2004.

[66] M. Magdon-Ismail, A. F. Atiya, A. Pratap, and Y. S. Abu-Mostafa, "On the maximum drawdown of a brownian motion," *Journal of applied probability*, vol. 41, no. 1, pp. 147–161, 2004.

[67] M. Campello, J. R. Graham, and C. R. Harvey, "The real effects of financial constraints: Evidence from a financial crisis," *Journal of financial Economics*, vol. 97, no. 3, pp. 470–487, 2010.