Applied Soft Computing 108 (2021) 107461

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Data selection to avoid overfitting for foreign exchange intraday trading with machine learning

Yuan-Long Peng [*], Wei-Po Lee

Department of Information Management, National Sun Yat-sen University, Kaohsiung 80424, Taiwan

ARTICLE INFO

Article history:
Received 26 August 2020
Received in revised form 29 March 2021
Accepted 23 April 2021
Available online 8 May 2021

Keywords:
Machine learning
Algorithmic trading
Foreign exchange
Overfitting

ABSTRACT

Algorithmic trading requires tuning hyperparameters to fit the time series data; however, it often suffers from overfitting of data that can lead to loss of money in action. Further, only a few studies discuss how to select trading exchange pairs and frequencies in response to the fitness of machine learning models. To cope with these problems, we developed a log-distance path loss model (to measure and reduce the overfitting in data modeling and determine exchange pairs and frequencies effectively. We conducted several experiments for different metrics using several influential factors such as machine learning models, learning objectives, trading strategies, and hyperparameter turning cases to validate the proposed approach. The obtained results indicate that the proposed metric is significantly superior to other methods in terms of accuracy, in-sample return (i.e., return of training data), and F1-score. Thus, using our path loss metric to guide data modeling, we provide a method to deal with the overfitting problem and yield positive trading returns.

1. Introduction

Algorithmic trading has been a popular topic for many decades owing to its many advantages over human trading including high speed and the ability to not be affected by emotions [1,2]. Most common algorithmic trading methods implement technical analyses [3,4] which suggest indicators of long and short strategies. However, the efficient-market hypothesis states that technical analysis is not helpful for making profits and it can concisely explain why it is very difficult to make money from trading or making investments. In a foreign exchange market, exchange quotes change promptly in response to rapidly updated information; therefore, there is a need to develop a more effective approach.

Machine learning can induce rules by iteratively reducing gaps between predicted and actual values. Many machine-learning methods have been applied to FX applications; often, the focus is on modeling performance that measures the accumulated discrepancy between the collected time series data and those produced by the inferred model over time [5,6]. Indeed, the objective of trading is undoubtedly a return [7]. However, long and short positions have different returns, and trading needs to be stopped when the returns become negative. Furthermore, maximizing the return could result in the algorithm (e.g., the model) fitting to the noise of the data instead of the hidden patterns. Therefore, we propose to convert the returns of long and short positions into a single value for the predictive model; this will simultaneously filter the spike return and retain the relative extent of the long and short position returns.

In addition to the computational techniques, two important issues need to be addressed in the data modeling process for automated trading: The first issue is overfitting. The relevant hyperparameters must be tuned repeatedly to fit the data to a machine learning model for optimization [8–11]. The model can overfit the training data, and consequently, the predictive performance of the model declines when the inferred model is applied to the test data [12,13]. Overfitting can be overcome using a number of methods including the addition of the L1/L2 regularization to the loss function [14,15] or of noise to the original data during the optimization process [16]. This method involves manually tuning the hyperparameters to eliminate noise; there is still a requirement for a more efficient and effective method.

The second issue is identifying how to select a currency pair and frequency that can couple with the data and the algorithm to obtain the best performance. In previous studies, different methods for time-series prediction performed well [6,17–20]. However, the obtained "model returns" only proved that the model worked for some specific data frames selected for the modeling. The trading frequency and currency pair need to be assigned in advance [7,21]; however, there is no general rule to determine the values for trading; we must depend on repeated human trials. To the best of our knowledge, no research has been published on choosing the appropriate trading FX pairs

* Corresponding author.
E-mail address: d044020002@student.nsysu.edu.tw (Y.-L. Peng).

https://doi.org/10.1016/j.asoc.2021.107461
1568-4946/

and trading frequency that can work with the machine learning algorithm.

To resolve the above two issues, we develop a log-distance path loss model (hereafter, path loss) with an overfitting measurement. Path loss has been widely used to measure signal loss within environments [22,23]. In this study, we use the analogy of different combinations of trading frequencies and currency pairs as the transmission environment and define a new path loss metric to guide the data modeling. Path loss is required to measure the log-distance between the normalized in-sample return and the profitable ratio with accuracy as that of the exponent. With these specially designed characteristics, our path loss approach can help determine the trading frequency and currency pair to further distinguish less overfitting data from original data. That is, our approach can overcome the overfitting problem and simultaneously lead to more stable returns for algorithmic trading.

We conducted several experiments that compared the proposed path loss metric with several popular metrics (e.g., accuracy, in-sample return, and F1-score) based on the following learning objectives, two trading strategies, and four machine learning models (neural networks, random forest, support vector machine/support vector regression (SVM/SVR) and eXtreme gradient boosting (XGBoost)) to verify the proposed approach. Our results prove that trading combinations that employ path loss as the metric outperform those combinations that use metrics of accuracy and in-sample returns. We fine-tune hyperparameters for the trading data (i.e., currency pair and trading frequency) selected using metrics of path loss, accuracy, in-sample return, and F1-score to further enhance the trading performance. Our results prove that the path loss metric can obtain positive returns when coupled with the objective of regression and the trading strategy of holding the position until switching. The process applied in this study is illustrated in Fig. 1.

2. Related work

Thus far, various methods have been proposed for algorithmic trading. One of the simple methods for algorithmic trading is a time-series prediction. The principle is that if the predicted $price_t$ is higher than $price_{t-1}$, then the long position is taken and vice versa. The popular methods employed for time-series prediction include autoregression (AR), moving average (MA), and autoregressive integrated moving average (ARIMA) [24,25]. The AR models the time-series data related to the previous data, and MA removes the high-frequency noise. Further, ARIMA combines MA and AR with stationary data; the autoregressive conditional heteroskedasticity model and generalized autoregressive conditional heteroskedasticity are used to model the volatility of the time-series data [26,27]. The above models are widely employed in financial research to model data and verify that time-series data are related to their own lagged values. Neural networks [28,29] and SVM/SVR [30–32] have had tremendous success for nonlinear data predictions; however, these have the problem of overfitting in out-of-sample data trading.

In addition to a single computational model, ensemble learning provides a useful prediction alternative. It includes a group of predictive models where the output is the synthesis of various models. In general, a group of models that work together outperform single models. To date, ensemble learning has been widely applied for predicting foreign exchange, and it has been shown to work well [29,33–35]. In this research, we adopted two ensemble machine learning methods (random forest and XGBoost) because they demonstrated good performances. XGBoost has won several data competitions, and therefore, it is considered a state-of-the-art method.

The selection of appropriate data features plays an important role in data modeling. The technical indicator is composed of buy and sell signals, and thus, it is widely used as a data feature for different computational methods in the trading and investment. Certain studies showed that technical indicators are positively related to returns [3,21,36]; however, the transaction costs may offset meager profits. From a machine learning perspective, a return is a natural reward for reinforcement learning. Reinforcement trading requires rewards, status, and actions for policy training [21]. The status includes information for the algorithms such as volume, price, or technical indicators; the actions of reinforcement trading include long, short, and holding positions.

Overfitting and the selection of currency pairs and frequency are two issues that need to be addressed in addition to computational techniques for performing algorithmic trading. Overfitting refers to the fact that the model fits noise rather than the patterns [37]. Bailey et al. [38] proposed a probability method to measure overfitting quantitatively; however, their method requires cross-validation to project probability when sequential patterns are undermined. Harvey and Liu adopted the Sharpe ratio, which is a metric that adjusts the return by its volatility to conduct statistical testing. A continuous positive return results in an inferior Sharpe ratio performance that may not result from overfitting [39]. Further, Carr and Prado introduced a method to generate out-of-sample data by modeling financial variables [40]. This implies that the measurement of overfitting depends on the quality of the model. Out-of-sample returns is a good measurement of overfitting that implies hyperparameters were fine-tuned based on the out-of-sample return. Therefore, these out-of-sample data are part of the in-sample data [41]. Other methods such as L1/L2 regularization and tree pruning can avoid overfitting, but these methods still require considerable human effort to repeatedly tune the hyperparameters. The process of fine-tuning hyperparameters involves selecting the best result within a range of hyperparameters for SVM and XGBoost. For the neural network and random forest, we trained a set of models with different hyperparameters and selected the best one.

The selection of currency pairs and frequency is the second issue with algorithmic trading. To resolve this issue, several popular metrics of in-sample performance including the root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), in-sample return, and accuracy can be used for the selection [42]. In this study, we focus on trading; therefore, reducing the RMSE, MAE, and MAPE will unnecessarily lead to a positive return. Further, the return will be positive only if the model predicts the right position (long/short). Therefore, only accuracy, in-sample return, and F1-score are suitable for our research. We adopted four metrics (accuracy, in-sample return, F1-score, and a specially designed path loss model) to measure in-sample performance. The path loss model is commonly used to measure the signal loss caused by the environment.

$$path\ loss \propto \gamma log(\frac{d}{d_0}) \tag{1}$$

Path loss is expressed as the log-distance of d divided by the base distance d_0, which is then multiplied by the exponential factor γ. According to Eq. (1), we develop a new model that can measure and overcome overfitting caused by noise. Further, our model can select the most appropriate exchange pairs and frequencies for trading.

3. Method

In this section, we present the proposed approach for automated trading and provide relevant details step-by-step. In Section 3.1, we first define the returns of the long and short

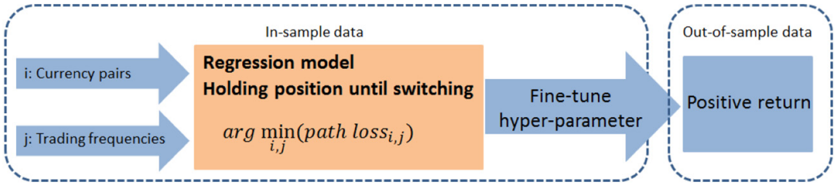


Fig. 1. Abstract process of this research.

positions and then use them as the basis to define the trading objectives from two perspectives: regression and classification. We then describe the two trading strategies with different assumptions for various conditions. One strategy is to hold the position for one period, and the other is to hold the position until switching. Both trading strategies take a long position and a short position based on the objectives. In Section 3.2, we define three metrics that include accuracy, in-sample return, and path loss to select the trading currency pair and frequency. Using conditions configured from the above factors, we depict the four machine-learning models and the input features used in this research in Section 3.3. In Section 3.4, we describe how the proposed approach is verified using a moving window.

3.1. Objectives

3.1.1. Returns of long/short positions

The return of long position (RLP) and return of short position (RSP) are respectively defined as

$$RLP_{t+1,i,j} \equiv \frac{Bid\ price_{t+1,i,j}}{Ask\ price_{t,i,j}} - 1 \tag{2}$$

$$RSP_{t+1,i,j} \equiv \frac{Bid\ price_{t,i,j}}{Ask\ price_{t+1,i,j}} - 1 \tag{3}$$

where t refers to time; i refers to the currency pair (one of the seven major trading currency pairs: AUD/USD, EUR/USD, GBP/USD, NZD/USD, USD/CAD, USD/CHF, and USD/JPY), and j refers to trading frequency, which includes candidates ranging from ten min to one d. This results in 144 trading intraday frequencies. The bid price and ask price refer to the quotation of the deal for buying and selling a currency, respectively. RLPs and RSPs are used to define the objective for the regression, profitable ratio, and trading strategy as follows:

3.1.2. Objectives of the regression and classification models

Our automated trading strategy is to place orders on a given frequency; therefore, slippage has a neutral effect on returns. Further, most foreign exchange traders do not charge commissions, and in our data source (Dukascopy), only a meager commission was charged. Therefore, the commission can be ignored and only the spread is considered as the transaction cost.

To define the objective by simultaneously considering RSP, RLP, and the spread, we project them onto the same plan. In Fig. 2, the x-axis represents the RSP, and the y-axis represents the RLP. Trading records located in Quadrant II and Quadrant IV imply that the long position and the short position are profitable, respectively. Further, another trading record located in Quadrant III indicates that both long and short positions are unprofitable. Thus, the objective of the regression is to learn the angle ($\pm\omega$) between the vector (1,1) and the vector of the long and short position returns (i.e., $\pm$ in the figure); this is equated as

$$angle \equiv \omega \times S = cos^{-1}(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}) \times S \tag{4}$$

where $\vec{a} = (1, 1)$, $\vec{b} = (RLP, RSP)$ and $S = sign(RLP - RSP)$.

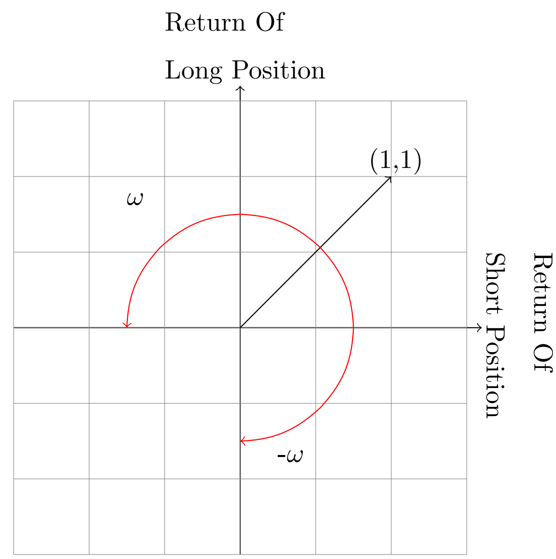


Fig. 2. Objective of the regression and classification model. For trading belonging to the vector space constituted by RLP and RSP, the measurement of the angle between the vector (1, 1) and the vector of RLP and RSP is defined as the objective of regression. Further, the objective of classification is to correctly classify data into three classes: *Long*, *Short* and *Stop trading* corresponding to the intervals of angles $0 < \omega < 3/4\pi$, $0 > \omega > -3/4\pi$, and $3/4\pi < \omega < 5/4\pi$, respectively.

We designed the objective as a classification problem to maximize accuracy based on another perspective of algorithmic trading. Classes of FX trading with the classification were $\{Long, Short, Stop\ trading\}$, which corresponded to angles $\{0 < \omega < 3/4\pi, 0 > \omega > -3/4\pi, 3/4\pi < \omega < 5/4\pi\}$ respectively. The trading cost is the spread of the bid and ask prices in this research, and it differentiates the RLPs and RSPs. The objectives of classification and regression models are both composed of RLP and RSP, and this means the trading cost is considered in the machine learning models.

3.1.3. Trading strategy

We adopted two trading strategies: holding position for one period and holding position until switching. Both trading strategies consider their own assumptions. The holding position for one period assumes that all data are independent and identical to the same distribution. The holding position until switching involves taking the long/short/close position when the algorithm recommends a position different from the previous one. Trading is assumed to be a Markov decision process (MDP) because each decision is conditioned on the previous position.

The first trading strategy (i.e., holding position for one period) takes the position based on the objective defined in Eq. (5). That is, if $0 < \omega < 3/4\pi$ the strategy takes the long position with $RLP_{t,i,j}$; if $-3/4\pi < \omega < 0$ the strategy takes the short position with $RSP_{t,i,j}$; and otherwise, no action is taken with $Return_{t,i,j} = 0$.

Table 1
Transitional position matrix is the currency transiting from t1 to t, and it is denoted as $Transaction_{i,j}(t-1,t)$.

	$Long_t$	$Short_t$	$Stop_t$
$Long_{t-1}$	0	$\frac{(Bid\ price_{t,i,j})^2}{1}$	$\frac{(Bid\ price_{t,i,j})}{1}$
$Short_{t-1}$	$\frac{1}{(Ask\ price_{t,i,j})^2}$	0	$\frac{1}{(Ask\ price_{t,i,j})}$
$Stop_{t-1}$	$\frac{1}{(Ask\ price_{t,i,j})}$	$\frac{(Bid\ price_{t,i,j})}{1}$	0

$$Return_{t,i,j} = \begin{cases} RLP_{t,i,j}, \text{ if } & 0 < \omega_{t,i,j} < 3/4\pi \\ RSP_{t,i,j}, \text{ if } & -3/4\pi < \omega_{t,i,j} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

For this strategy, in-sample and out-of-sample returns are accumulations of the compounded $Return_{t,i,j}$ defined as

$$Return_{i,j} = \prod_{t=1}^{n}(Return_{t,i,j}+1) - 1 \tag{6}$$

The second trading strategy – holding position until switching – conforms to the MDP, and we list the transitional condition in Eq. (7). For the position transitioning from t1 to t, if $0 < \omega < 3/4\pi$, the strategy takes a long position; if $-3/4pi < \omega < 0$, the strategy takes a short position; otherwise, the holding position is taken. Table 1 summarizes the transactions made while transitioning from $position_{t-1}$ to $position_t$. For this strategy, the return of the holding position until switching is the product of the transactions, as defined in Eq. (8); all positions should be closed at the end of trading.

$$Position_{t,i,j} = \begin{cases} Long\ Position_{t,i,j}, \text{ if } & 0 < \omega_{t,i,j} < 3/4\pi \\ Short\ Position_{t,i,j}, \text{ if } & -3/4\pi < \omega_{t,i,j} < 0 \\ Holding\ Position_{t,i,j}, & \text{otherwise} \end{cases} \tag{7}$$

$$Return_{i,j} = \prod_{t=2}^{n} Transaction_{i,j}(t-1,t) - 1 \tag{8}$$

Further, we include the buy-and-hold training strategy as the baseline, and we assume that the trader has unlimited cash. For other trading strategies, we assume the trader has a unit of money in the beginning. When the return reaches −100%, the trader loses all the money. In addition, we did not consider leverage.

3.2. Selection of currency pair and trading frequency

We emphasize the importance of selecting trading data. In this subsection, we describe how we develop and exploit a path loss model as a metric to select the currency pair and trading frequency that determine the most appropriate data for modeling. Further, the metrics of accuracy and in-sample return are included as baselines for comparison. The metric of the in-sample return is discussed in Eqs. (6) and (8). Here, we only depict accuracy and path loss. With these metric definitions, we define the corresponding rules for selecting currency pairs and trading frequencies.

3.2.1. Accuracy

Accuracy is a popular metric for evaluating the performance of a classification problem. Therefore, to measure the accuracy of the regression model, we turn the prediction of regression into categorical results. Table 2 lists the confusion matrix that contrasts the predictive results with the ground truth in the classification. Based on this matrix, the accuracy is defined as the ratio of the sum of the diagonal entries to the sum of all entries. Accuracy is a component of the path-loss model and it is given as

Table 2
Confusion matrix includes the ground truth and prediction of the long position (Long), short position (Short), and stop trading (Stop).

		Ground truth		
		Long	Short	Stop
Prediction	Long	a_1	a_2	a_3
	Short	a_4	a_5	a_6
	Stop	a_7	a_8	a_9

$$Accuracy = \frac{a_1 + a_5 + a_9}{\sum_{i=1}^{9} a_i}. \tag{9}$$

where a_i refers to the amount of prediction related to the ground truth for each category.

3.2.2. Path loss

In our approach, we adopted the path loss concept from the log-distance of the path loss model. Here, we consider trading data as the medium and use the analogy of the overfitting phenomenon for the path loss accumulated over the trading period. We replace the terms γ, d_0, and d used in the original model with accuracy, profitable ratio, and normalized in-sample return, respectively. First, we describe the profitable ratio and normalized in-sample return, and then, we use these to define the path loss.

Different trading frequencies indicate different numbers of transactions. If each transaction consumes or gains a small amount of capital (e.g., 0.01%) for high-frequency trading, the total capital can be lost or doubled during the period of data backtesting (but not for low-frequency trading). Therefore, in-sample returns should be normalized to the same scale. A normalized in-sample return (NISR), which is a one-minute return, is defined as

$$NISR_{i,j} = \sqrt[1440]{(IS\ Return_{i,j}+1)} \tag{10}$$

The profitable ratio measures the proportion of price fluctuations that cover the spread; this is defined as

$$\#\ of\ profitable\ position_{i,j} = \sum_{t=1}^{n} \frac{max(RLP_{t,i,j},\ RSP_{t,i,j},\ 0)}{max(RLP_{t,i,j},\ RSP_{t,i,j})} \tag{11}$$

$$Profitable\ ratio_{i,j} \equiv \frac{\#\ of\ profitable\ position_{i,j}}{n} \tag{12}$$

If the profitable ratio is 1, the machine learning model must learn only the sign of the objective. In other words, the data are not located in quadrant III, as indicated in Fig. 2.

A high NISR implies a high tendency for overfitting. Hence, the path loss measures the log distance between the NISR and the profitable ratio, and then, it multiplies by the accuracy. A low path loss implies that the returns stem from patterns instead of noise. A high NISR implies a high tendency for overfitting. Therefore, our path loss model first measures the log-distance between the NISR and the profitable ratio. Subsequently, it multiplies this distance with accuracy. The overall model is defined by

$$Path\ loss_{i,j} = accuracy_{i,j} log(\frac{NISR_{i,j}}{profitable\ ratio_{i,j}}) \tag{13}$$

With this measurement, the trained models can overcome the overfitting problem and simultaneously obtain better results.

In summary, the path loss factors into model performances (accuracy and NISR), data characteristics (profitable ratio), and the interaction of both (log of NISR divided by profitable ratio).

3.2.3. Selection of trading data

We adopted three metrics to measure in-sample performance: accuracy, in-sample return, and path loss. Accuracy measures the correctness of a model in classifying data into three classes: taking a long position, short position, or stopping trading. In-sample returns measure the compounded returns of the in-sample data. Path loss measures the degree of model fitting to the noise. These metrics are used to select the best currency pair and trading frequency through the processes indicated by

$$arg\ \max_{i,j}(accuracy_{i,j}) \tag{14}$$

$$arg\ \max_{i,j}(IS\ return_{i,j}) \tag{15}$$

$$arg\ \min_{i,j}(path\ loss_{i,j}) \tag{16}$$

where i and j represent the currency pair and trading frequency, respectively. Currency pair i and trading frequency j obtained from the data frame (Section 3.4) are the best choices for trading. As shown in Eqs. (14)–(16), both the metrics of accuracy and in-sample return require a maximum value; the path loss requires a minimum value because it measures overfitting.

3.3. Training and testing of the machine learning model

Based on the above definitions and descriptions of the critical factors in trading, we introduce the four machine learning models and input features used to validate the metrics. In the evaluation, the out-of-sample return was calculated using a moving window.

3.3.1. Data

For modeling and testing, we collected per-minute data from Dukascopy [43] for the years 2007 to 2020. The data included seven major trading currency pairs: AUD/USD, EUR/USD, GBP/USD, NZD/USD, USD/CAD, USD/CHF, and USD/JPY.

3.3.2. Input features

The six input features used for our machine learning method are the spreads, changes in bid prices, changes in ask prices, differences in bid and ask volumes, volatility of bid prices, and volatility of ask prices. These are defined as

$$SP_{t,i,j} \equiv close\ price\ of\ bid_{t,i,j} - close\ price\ of\ ask_{t,i,j} \tag{17}$$

$$CBP_{t,i,j} \equiv close\ price\ of\ bid_{t,i,j} - close\ price\ of\ bid_{t-1,i,j} \tag{18}$$

$$CAP_{t,i,j} \equiv close\ price\ of\ ask_{t,i,j} - close\ price\ of\ ask_{t-1,i,j} \tag{19}$$

$$DBAV_{t,i,j} \equiv volume\ of\ bid_{t,i,j} - volume\ of\ ask_{t,i,j} \tag{20}$$

$$VBP_{t,i,j} \equiv high\ price\ of\ bid_{t,i,j} - low\ price\ of\ bid_{t,i,j} \tag{21}$$

$$VAP_{t,i,j} \equiv high\ price\ of\ ask_{t,i,j} - low\ price\ of\ ask_{t,i,j} \tag{22}$$

The spread (SP) is positively related to volume and negatively related to the effect of information [44]. The changes in the bid price and ask price imply trading initiated by the buyer or seller [45,46]. Because volatility has been proved to be related to returns [47], the volatility of the bid price and that of the ask price are included. The differences in the bid and ask volumes are used to measure the volume imbalance, which is related to price fluctuations [48].

3.3.3. Machine learning model

Four popular machine learning models are adopted for modeling in this study: SVM/SVR, random forest, XGBoost, and a neural network. They are used to work with the objectives of regression and classification. The performance of the modeling process is further enhanced by fine-tuning the hyperparameters involved. However, it is not practical to fine-tune all trading data (i.e., all configurations of parameters related to critical factors defined in the above sections) because it requires considerable human effort.

Our major focus is on the selection of trading currency pairs and frequencies, and therefore, we only fine-tune the hyperparameters for the trading data selected by the metrics. Therefore, trading is performed in two phases: determining the best hyperparameters and fine-tuning them. The determination of the best hyperparameters is an inefficient and time-consuming process because the combination of the currency pair and trading frequency for each time period is 1008. Using the default setting is a conservative approach to select currency pairs and trading frequencies for the next step. We used the default settings of the R library for the hyperparameters as follows: the kernel of the SVM and SVR is the radial basis function, the tree number for the random forest is 500, the maximum number of boosting iterations for XGBoost is 5, and the dimensions of the three layers of neural networks are (6,3,1) and (6,3,3) for regression and classification, respectively. We only fine-tuned the hyperparameters for each currency pair and frequency selected with each metric to train the model.

3.4. Moving window

The collected data were divided into 52 data frames to perform modeling; each frame included in-sample and out-of-sample data. The moving window technique was employed to access the time-series data in sequence. All input data were scaled using the mean and variance of the in-sample data. Further, the machine learning models were trained using in-sample data with input features and the objectives stated above (Sections 3.3.2 and 3.3.3). The in-sample returns are the compounded returns of the in-sample data for a given set of currency pairs and trading frequency. The 52 data frames represent the annual experimental data (from January 2007 to September 2020); we used nine months of data for training and the other three months for testing.

Table 3 provides a walk-through example of using the moving-window technique to perform trading. In this example, the XGBoost method is adopted to select the currency pair and trading frequency. We conducted trials for both types of objectives (regression and classification), and the trading strategy of the holding position for one period was used. The best results for the metrics of accuracy, in-sample return, and path loss are summarized in Table 3. For each case, the corresponding trading frequency (i.e., the period in the table), currency pair, and out-of-sample returns are listed. Further, the geometric average return (i.e., "Geo. Avg.") are shown in the last row of the table. In this example, geometric average returns obtained by the metric of the path loss are better than those obtained by the metrics of accuracy and in-sample return. Further, these results show that it is possible to obtain a positive geometric average return by taking the perspective of classification and using the path loss metric to construct models. This is very important for algorithmic trading.

4. Results and discussions

We conducted a series of experiments to verify the proposed approach and presented the results. We evaluated different combinations of the critical factors analyzed in Section 3 including configurations of two learning objectives (regression and

Table 3
Example of XGBoost selecting currency pair and trading frequency (period) and its corresponding out-of-sample return. For the objective of regression and classification with the trading strategy of holding the position for one period, frequency (period) and currency pairs are selected using accuracy, in-sample return, and path loss. The geometric return of the out-of-sample returns are shown.

Regression												
	Accuracy				In-sample return				Path loss			
	Accuracy (Max)	Period (Mins)	Currency pair	Out-of-sample return	In-sample return (Max)	Period (Mins)	Currency pair	Out-of-sample return	Path loss (Min)	Period (Mins)	Currency pair	Out-of-sample return
200 701	85.20%	360	AUD/USD	19.07%	14 100.65%	950	NZD/USD	−52.88%	1.01E−04	1390	USD/JPY	−8.45%
200 704	91.34%	30	AUD/USD	−88.40%	27 280.73%	340	AUD/USD	1.45%	1.09E−04	1410	USD/JPY	−28.85%
200 707	89.87%	20	AUD/USD	−93.76%	48 196.25%	770	NZD/USD	−29.25%	1.39E−04	1140	USD/JPY	−24.87%
200 710	87.47%	70	EUR/USD	−55.93%	18 031.95%	600	AUD/USD	−18.16%	1.27E−04	1180	USD/JPY	50.86%
Geo. Avg.				−75.17%				−27.47%				−7.31%
Classification												
	Accuracy				In-sample return				Path loss			
	Accuracy (Max)	Period (Mins)	Currency pair	Out-of-sample return	In-sample return (Max)	Period (Mins)	Currency pair	Out-of-sample return	Path loss (Min)	Period (Mins)	Currency pair	Out-of-sample return
200 701	79.63%	710	AUD/USD	14.39%	15 403.55%	600	NZD/USD	−25.92%	9.52E−05	1390	USD/JPY	6.82%
200 704	83.56%	150	AUD/USD	−27.88%	36 611.16%	280	AUD/USD	−16.95%	1.14E−04	1410	USD/JPY	−6.21%
200 707	84.04%	240	AUD/USD	−22.19%	76 761.17%	190	AUD/USD	−46.55%	1.44E−04	1140	USD/JPY	−16.81%
200 710	80.07%	850	AUD/USD	−45.95%	37 221.49%	250	AUD/USD	−58.58%	1.38E−04	1180	USD/JPY	30.16%
Geo. Avg.				−23.25%				−39.25%				2.06%

classification), two trading strategies (holding position for one period and holding position until switching), three metrics (in-sample return, accuracy, and path loss) employed to select the trading data (frequency and currency pair), and four machine learning models (SVM/SVR, random forest, XGBoost, and neural network) adopted for modeling. Further, we show a comparison of the results for the trials with and without fine-tuning the hyperparameters. To observe variations in model behavior during trading, we plot the cumulative out-of-sample returns over time in Figs. 3–6. Table 4 summarizes the compounded out-of-sample returns corresponding to Figs. 3–6. Further, the results in green in Table 4 are positive, and those in red are negative; those in bold text indicate the best in three metrics with the same configuration of the model, trading strategy, objective, and hyperparameter settings. We discuss the results from the four perspectives of algorithmic trading to further investigate the effects of different factors. These four perspectives include the metrics, trading strategy, fine-tuning hyperparameters, and machine learning model.

4.1. Performance of metrics in selecting trading currency pairs and frequencies

We developed three metrics to select currency pairs and frequencies for trading: path loss, accuracy, and in-sample return. Table 4 lists the compounded out-of-sample returns and the best out-of-sample returns among the three metrics presented in bold. From the metric evaluation perspective, these results confirm that path loss can deliver the best performance and is the best metric. Further, in the experiments, the path loss metric can lead to positive out-of-sample returns when used with the objective of regression for the strategy of holding position until switching and fine-tuning hyperparameters. While considering algorithmic trading as a regression or classification problem, path loss is found to be the best for both objectives. The path loss metric outperforms other metrics because it considers the profitability of the data (profitable ratio) and the fitness of the models (accuracy and NISR).

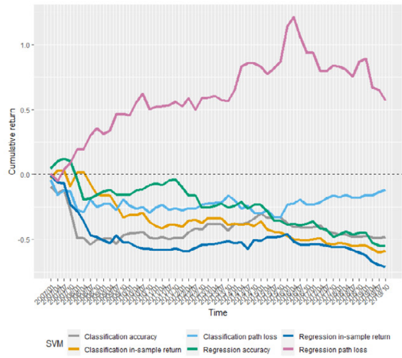
4.2. Comparison of trading strategies

We conducted several sets of experiments and categorized and compared the results to evaluate the two trading strategies

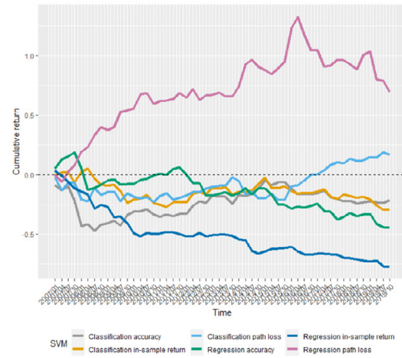
Table 4
Compounded return for all combinations of trading strategies, machine learning models, objectives, and without and with fine-tuning hyperparameters. The returns in bold are the best cases among the three metrics. The details of parentheses and conclusions are discussed in Sections 4.2 and 4.4, respectively.

Method	Path loss			
	Regression	Classification	Regression	Classification
	Without fine-tuning		Fine-tuning	
NN	**−21.40%**	**−44.62%**	**−10.59%**	**−54.40%**
NN holding	**38.03%**	**20.86%**	**29.39%**	**28.15%**
RF	**10.09%**	**−28.54%**	**14.99%**	**−43.25%**
RF holding	**(−6.25%)**	**−6.10%**	**(11.15%)**	**(−51.77%)**
SV	**56.94%**	**−11.62%**	**49.28%**	**−43.43%**
SV holding	**69.24%**	**17.00%**	**60.51%**	**−32.20%**
XG	**−48.66%**	**15.19%**	**−5.25%**	**−51.62%**
XG holding	**−39.33%**	**15.60%**	**28.92%**	**−36.48%**
	Accuracy			
NN	−48.26%	−68.69%	−41.58%	−72.26%
NN holding	16.57%	−2.01%	−9.00%	−35.13%
RF	−37.24%	−100.00%	−41.36%	−100.00%
RF holding	(−39.57%)	−99.99%	−29.26%	−100.00%
SV	−55.21%	−48.19%	−55.40%	−52.36%
SV holding	−44.37%	−21.36%	−44.34%	−32.82%
XG	−100.00%	−97.58%	−100.00%	−97.95%
XG holding	−99.61%	−77.22%	−99.99%	−85.15%
	In-sample return			
NN	−71.20%	−67.12%	−71.50%	−69.90%
NN holding	−10.56%	−49.33%	−9.75%	−48.32%
RF	−100.00%	−57.21%	−100.00%	−75.26%
RF holding	−100.00%	−22.85%	−100.00%	−57.34%
SV	−71.06%	−58.72%	−73.85%	−72.87%
SV holding	(−77.21%)	−28.98%	(−77.66%)	−56.80%
XG	−93.01%	−97.72%	−88.35%	−97.68%
XG holding	−86.16%	−92.63%	−73.09%	−95.38%

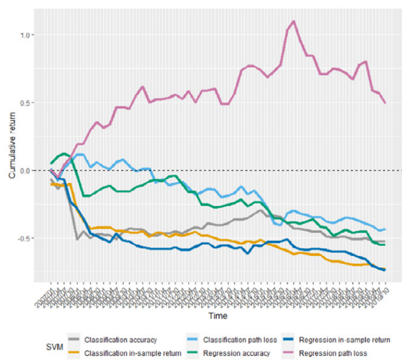
(i.e., holding position for one period and holding position until switching). Figs. 3–6 present the results obtained for the four machine learning methods. Figs. 3–6(a) and (c) show the results (without and with fine-tuning hyperparameters) for the first strategy (i.e., the holding position for one period), and (b) and (d) show the results for the second strategy (i.e., the holding position until the switching). The compound returns are summarized in Table 4 as indicated by the four learning methods without or with the "holding". The strategy of the holding position until



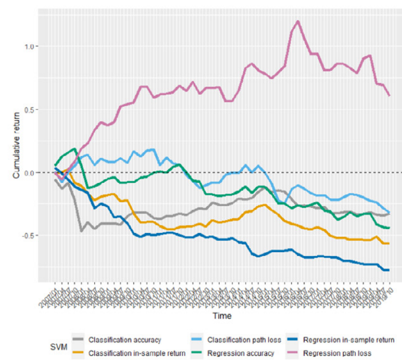
(a) SVM/SVR holding for one period



(b) SVM/SVR holding until switching



(c) SVM/SVR holding for one period and fine-tuned hyper-parameters



(d) SVM/SVR holding until switching and fine-tuned hyper-parameters

Fig. 3. SVM/SVR out-of-sample cumulative return with treatments of trading strategies, machine learning model, objective, and fine-tune hyperparameters. The metric of path loss can lead to positive return in (d), which also holds across all machine learning models.

switching is better than the strategy of the holding position for one period, excluding the seven cases labeled with parentheses. These comparisons suggest that FX trading should be conditioned on the previous position and conform to the MDP to obtain better results.

4.3. Effect of fine-tuning hyperparameters

We applied hyperparameters to the data specified by selecting the trading currency pair and frequency to investigate the effect of fine-tuning the hyperparameters. From the out-of-sample results listed in Table 4 (under the columns of without and with fine-tuning), we observe that only the trials configured by the metric of path loss and the objective of regression can lead to more positive returns after fine-tuning hyperparameters. All returns obtained with the other configurations are degraded after fine-tuning the hyperparameters. Comparison of the cumulative returns over time for trials with and without fine-tuning hyperparameters is shown in Figs. 3–6 (which corresponds to the four learning methods). For illustrative purposes, (a) and (b) represent variations in the cumulative returns for trading without fine-tuning, and (c) and (d) represent these variations with fine-tuning. As indicated, fine-tuning hyperparameters only works for configurations that use the metric of path loss and the objective of regression.

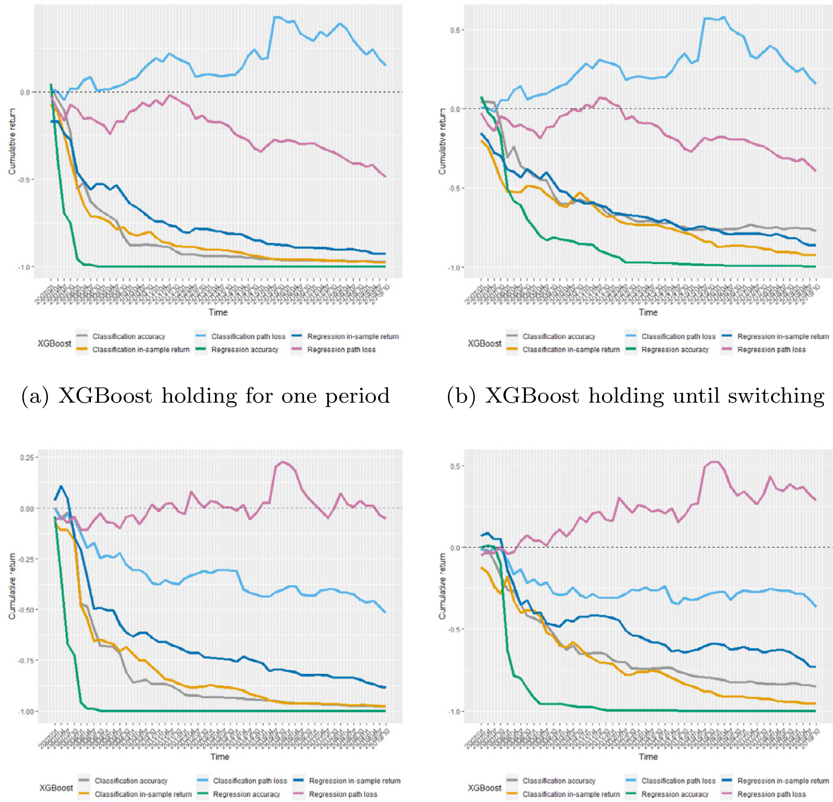
The path loss outperforms other metrics because it identifies trading frequency and currency pairs that are less likely to be affected by overfitting. Furthermore, the objective of regression contains the relative extent of long and short position returns, but the objective of classification does not. Therefore, fine-tuning parameters for the objective of regression enables the model to learn more information from the data.

4.4. Evaluation of machine learning models

The results listed in Table 4 show that none of the four machine learning models adopted can outperform the others in all scenarios. Some popular models (the ensemble learning methods of random forest and XGboost) produced an out-of-sample return of 100% when they were used with the metrics of accuracy and in-sample return. However, all learning models with the configuration of the metric of path loss, the objective of regression, and the trading strategy of holding until switching produce positive out-of-sample returns, which shows the generality and stability of the specific configuration on different machine learning models.

5. Further analysis

This study presents the returns of a linear model and a buy-and-hold trading strategy under the configuration of currency pair EUR/USD and trading frequency of 10 min to investigate whether a simple model can surpass our proposed method. To demonstrate the performance of popular approaches to deal with overfitting, a set of experiments was conducted wherein the neural network model was employed with L1/L2 regularization and dropout. Further, the mean and median of the losses of the in-sample and out-of-sample data are listed to observe whether the low difference between the in-sample and out-of-sample losses implies a low overfitting and a positive out-of-sample return.



(a) XGBoost holding for one period

(b) XGBoost holding until switching

(c) XGBoost holding for one period and fine-tuned hyper-parameters

(d) XGBoost holding until switching and fine-tuned hyper-parameters

Fig. 4. XGBoost out-of-sample cumulative return with treatments for trading strategies, machine learning model, objective, and fine-tuned hyperparameters. The metric of path loss can lead to positive return in (d), which also holds across all machine learning models.

5.1. Baselines

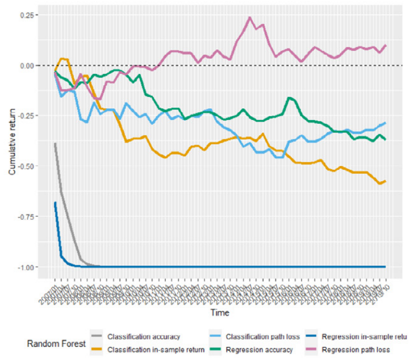
As baselines, this study adopted the most common currency pair ERU/USD with a trading frequency of 10 min to perform a naive buy-and-hold trading strategy and a linear model to learn the six features. Popular measures such as L1/L2 regularization and dropout were employed to deal with overfitting. The results in Fig. 7 show that the out-of-sample returns of the linear model converged to 100% and the neural networks of the regression model and classification model with L1/ L2 regularization and dropout. Although the buy-and-hold trading strategy suffered a minor loss, it required the assumption of unlimited capital that was not in line with the two trading strategies of this study.

5.2. F1-score

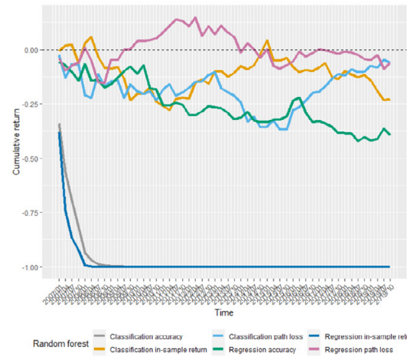
The F1-score is the harmonic of precision and recall, which is a more objective metric than accuracy. Table 5 presents the results of the F1-score and the path loss with the F1-score (to replace the original accuracy). The performance metric can obtain better out-of-sample returns using the F1-score instead of using in-sample returns and accuracy; five positive returns are obtained (Table 5). Compared with the path loss with accuracy, the path loss with the F1-score had the same number of positive returns for the configuration without fine-tuning the hyperparameters, whereas the path loss with the F1-score underperformed the path loss with the accuracy for configuring fine-tuning hyperparameters. The F1-score is a good metric; however, the path loss with accuracy is better than the F1-score in terms of the configuration of fine-tuning the hyperparameters. According to the result that the number of positive returns obtained using accuracy (in Table 4) is smaller than those obtained using the F1-score (in Table 5), the accuracy is confirmed to be more susceptible to overfitting than the F1-score. The path loss is designed to measure the overfitting; the minimization of accuracy in the path loss metric is to employed to minimize the chance of overfitting. Therefore, the path loss with accuracy is a better choice than that with the F1-score.

Table 5
Compounded returns of the path loss with the F1-score and of the F1-score.

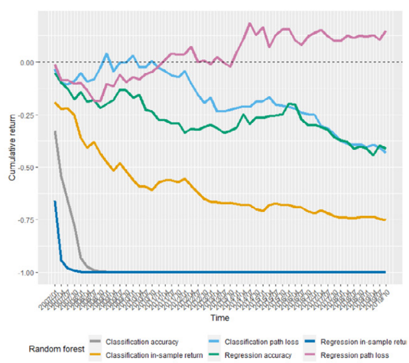
Method	Path loss with F1-score			
	Regres.	Classi.	Regres.	Classi.
	Without fine-tuning		Fine-tuning	
NN	**6.72%**	**−14.12%**	**−15.94%**	**−21.09%**
NN holding	**−16.45%**	**−21.43%**	**−16.14%**	**−4.71%**
RF	**−9.68%**	**−11.62%**	**−15.98%**	**−12.33%**
RF holding	**−18.82%**	**16.43%**	**−15.94%**	**−50.91%**
SV	**41.84%**	**−12.12%**	**34.91%**	**−45.20%**
SV holding	**66.57%**	**17.00%**	**57.99%**	**−32.21%**
XG	**−12.80%**	**33.86%**	**−6.14%**	**14.15%**
XG holding	**1.13%**	**51.32%**	**38.22%**	**18.84%**
	F1-score			
NN	−0.34%	−59.42%	−22.88%	−49.23%
NN holding	−20.70%	−8.39%	−17.42%	−28.05%
RF	−29.88%	−100.00%	−49.78%	−100.00%
RF holding	−23.03%	−100.00%	−29.80%	−100.00%
SV	−58.13%	−13.92%	−58.73%	−6.01%
SV holding	−46.95%	39.79%	−46.93%	51.25%
XG	−31.31%	61.70%	−64.08%	7.81%
XG holding	−6.48%	66.49%	−34.01%	−7.69%



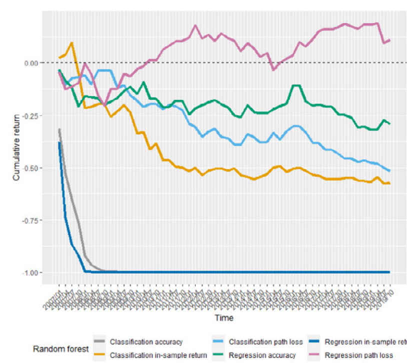
(a) Random forest holding for one period



(b) Random forest holding until switching



(c) Random forest holding for one period and fine-tuned hyper-parameters



(d) Random forest holding until switching and fine-tuned hyper-parameters

Fig. 5. Random forest out-of-sample cumulative return with treatments of trading strategies, machine learning model, objective, and fine-tuned hyperparameters. The metric of path loss can lead to positive return in (d), which also holds across all machine learning models.

5.3. Loss between in-sample and out-of-sample data

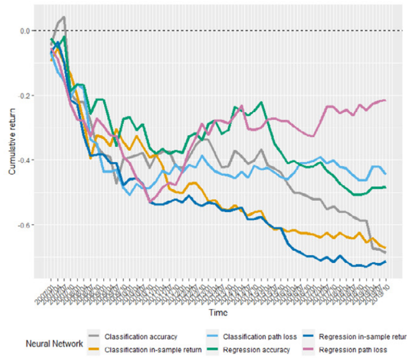
The trick to stopping early is based on the assumption that the model learns some patterns that are not included in the validation data when the validation loss is higher than the training loss. The low difference between the in-sample loss and out-of-sample loss implies a low chance of overfitting. Table 6 shows the means and medians of the in-sample and out-of-sample losses obtained by the neural network and XGBoost models with the trading strategy of holding the position until switching. The lowest difference between the in-sample and out-of-sample was based on accuracy; however, it did not have a positive out-of-sample return, excluding one case where neural networks were used for the objective of regression. Further, the path loss had the lowest out-of-sample loss because it can select trading data with a high profitable ratio. Thus, the data included relatively fewer stop positions. In other words, the data contained more long and short positions, which means that trading data are easier to manipulate than those with low profitability ratios. The path loss metric is composed of the performance of the model and the profitability of data. Therefore, the path loss metric can select a model that has truly learned the data patterns. The assumption of stopping early did not lead to a positive out-of-sample return. For random forest and SVM, neither packages in R provided support to output the out-of-sample loss. The training of the random forest model considers the out-of-bag error, which is similar to the out-of-sample data. Therefore, the results of the random forest were the minimum in-the-bag error and out-of-bag error.

5.4. Rationale for using the metric of path loss

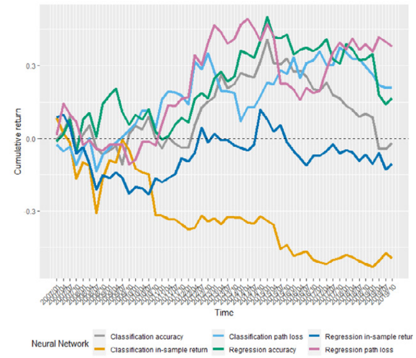
The highest trading frequency in our research was 10 min, and therefore, the normalized in-sample return must be close to 1. The log function can measure the distance between the normalized in-sample return and profitable ratio. A low log-distance implies a low chance of overfitting because high in-sample returns likely result in loss of out-of-sample data, as summarized in Table 4. Under the same condition of the log-distance, the low accuracy implies a low probability of overfitting because Table 4 indicates that high accuracy leads to negative out-of-sample returns. For example, consider two cases with similar profitable ratios of 99%, normalized in-sample returns of 102% and 101%, and similar accuracies of 97%, respectively. Here, the latter case of the trading frequency and currency pair is selected because the lower log-distance is less likely to be over-fitted. As another example, consider two other cases with profitable ratios of 99%, normalized in-sample returns of 102%, and accuracies of 96% and 97%, respectively. Here, the former case of trading frequency and currency pairs is selected because the lower accuracy with the same log-distance is less likely to be over-fitted.

6. Conclusion

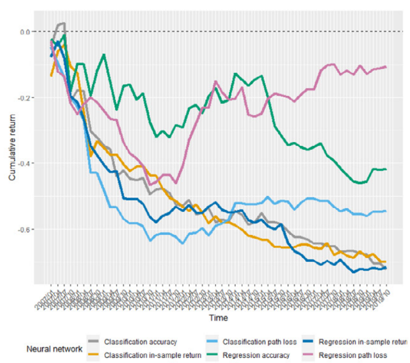
In this study, we focused on the importance of algorithmic trading and developed a log-distance path loss model as a metric to work with other influential factors for data modeling. The specific characteristics of the path loss model allow the metric to measure the performance affected by data noise and to



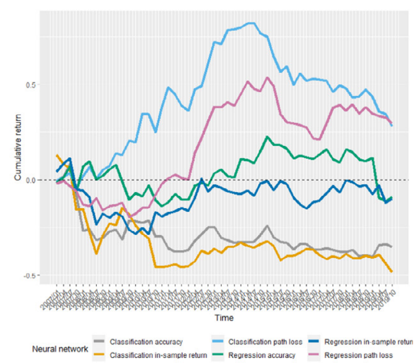
(a) Neural network holding for one period



(b) Neural network holding until switching



(c) Neural network holding for one period & fine-tuned hyper-parameters



(d) Neural network holding until switching & fine-tuned hyper-parameters

Fig. 6. Neural network out-of-sample cumulative return with treatments of trading strategies, machine learning model, objective, and fine-tuned hyperparameters. The metric of path loss can lead to positive return in (d), which holds across all machine learning models.

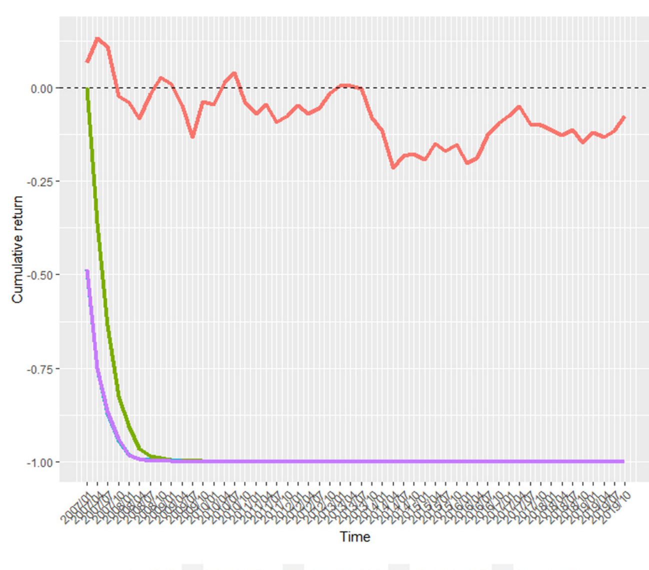


Fig. 7. Four baselines of buy-and-hold trading strategy, linear model, and neural network with L1/L2 regularization and dropout with regression, and a classification model.

derive a trading model that can overcome the overfitting problem. Further, the path loss metric was used to select trading exchange pairs and frequencies to maximize profit. Our experimental results demonstrated that the trading strategy of the holding position until switching was better than that of the holding position for one period, which indicates that FX trading is an MDP. Trials conducted for trading with specific configurations of the metric of path loss, objective of regression, strategy of holding position until switching, and fine-tuned hyperparameters can lead to positive out-of-sample returns regardless of the machine learning method employed. These results confirmed the effectiveness and efficiency of the proposed approach. The practitioners can pick up trading data with specific currency pairs and frequencies to fine-tune the model, and this can lead to a positive out-of-sample return.

CRediT authorship contribution statement

Yuan-Long Peng: Conducted the experiments, Wrote the manuscript. **Wei-Po Lee:** Consultation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by Ministry of Science and Technology of Taiwan under grant MOST-105-2221-E-110-077-MY2.

Table 6
Comparison of in-sample loss and out-of-sample loss of Neural network and XGBoost with holding position trading strategy.

Neural network						
Method	Regression					
	In-sample	Out-of-sample	In-sample	Out-of-sample	Profitable ratio	
	Mean		Median		Mean	Median
Return	2.763	11.430	2.751	3.051	97.39%	97.43%
Accuracy	2.915	2.971	2.846	2.969	96.49%	96.83%
F1-score	2.544	3.027	2.504	2.863	98.86%	99.05%
Path loss (Accuracy)	**2.648**	**2.755**	**2.610**	**2.747**	99.39%	100%
Path loss (F1-score)	**2.591**	**2.722**	**2.582**	**2.739**	100%	100%
	Classification					
Return	0.7791	0.8750	0.7710	0.8595	97.08%	97.44%
Accuracy	0.7932	0.8274	0.7939	0.8282	97.05%	97.42%
F1-score	0.7288	0.8299	0.7256	0.8235	98.38%	98.52%
Path loss (Accuracy)	**0.7024**	**0.7603**	**0.6864**	**0.7348**	98.77%	100%
Path loss (F1-score)	**0.7006**	**0.7617**	**0.6953**	**0.7473**	100%	100%
XGBoost						
	Regression					
Return	1.431	1.862	1.449	1.867	94.65%	95.19%
Accuracy	1.944	1.994	1.909	1.973	84.01%	86.55%
F1-score	0.8988	1.809	0.9084	1.809	99.45%	99.51%
Path loss (Accuracy)	**1.1190**	**1.759**	**1.1230**	**1.742**	100%	100%
Path loss (F1-score)	**1.1575**	**1.737**	**1.1588**	**1.723**	100%	100%
	Classification					
Return	0.2819	0.5455	0.2894	0.5452	91.30%	91.81%
Accuracy	0.3214	0.5238	0.3242	0.5302	94.57%	95.48%
F1-score	0.02834	0.4981	0.02985	0.5035	98.18%	98.58%
Path loss (Accuracy)	**0.12604**	**0.4928**	**0.12997**	**0.4893**	100%	100%
Path loss (F1-score)	**0.13191**	**0.4892**	**0.12978**	**0.4870**	100%	100%

References

[1] A.P. Chaboud, B. Chiquoine, E. Hjalmarsson, C. Vega, Rise of the machines: Algorithmic trading in the foreign exchange market, J. Finance 69 (2014) 2045–2084, The Journal of Finance.
[2] T. Hendershott, C.M. Jones, A.J. Menkveld, Does algorithmic trading improve liquidity?, J. Finance 66 (2011) 1–33.
[3] M.P. Taylor, H. Allen, The use of technical analysis in the foreign exchange market, J. Int. Money Financ. 11 (1992) 304–314.
[4] N. Taylor, The rise and fall of technical trading rule success, J. Bank. Financ. 40 (2014) 286–302.
[5] G. Sermpinis, C. Stasinakis, R. Rosillo, D. de la Fuente, European Exchange trading funds trading with locally weighted support vector regression, European J. Oper. Res. 258 (2017) 372–384.
[6] Y. Deng, F. Bao, R.Y.Z. Kong, Q. Dai, Deep direct reinforcement learning for financial signal representation and trading, IEEE Trans. Neural Netw. Learn. Syst. 28 (2017) 653–664.
[7] G. Sermpinis, C. Dunis, J. Laws, C. Stasinakis, Forecasting and trading the EUR/USD exchange rate with stochastic neural network combination and time-varying leverage, Decis. Support Syst. 54 (2012) 316–329.
[8] D. Anguita, S. Ridella, F. Rivieccio, R. Zunino, Automatic hyperparameter tuning for support vector machines, in: International Conference on Artificial Neural Networks, 2002, pp. 1345–1350.
[9] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, Neural Inf. Process. Syst. (2012) 2951–2959.
[10] R.G. Mantovani, A.L. Rossi, J. Vanschoren, B. Bischl, A.C. Carvalho, To tune or not to tune: recommending when to adjust SVM hyper-parameters via meta-learning, in: Proceedings of the International Joint Conference on Neural Networks, 2015, pp. 1–8.
[11] K. Duan, S.S. Keerthi, A.N. Poo, Evaluation of simple performance measures for tuning svm hyperparameters, Neurocomputing 51 (2003) 41–59.
[12] P.M. Domingos, A few useful things to know about machine learning, Commun. ACM 55 (2012) 78–87.
[13] J. Beckmann, R. Schüssler, Forecasting exchange rates under parameter and model uncertainty, J. Int. Money Financ. 60 (2016) 267–288.
[14] X. Luo, X. Chang, X. Ban, Extreme learning machine for regression and classification using L1-Norm and L2-Norm, in: In Proceedings of ELM-2014, Vol. 1, 2015, pp. 293–300.
[15] X. Luo, X. Chang, X. Ban, Regression and classification using extreme learning machine based on L1-norm and L2-norm, Neurocomputing 174 (2016) 179–186.
[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[17] L. Luo, X. Chen, Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction, Appl. Soft Comput. 13 (2013) 806–816.
[18] J.L. Wu, L.C. Yu, P.C. Chang, An intelligent stock trading system using comprehensive features, Appl. Soft Comput. 23 (2014) 39–50.
[19] X.D. Zhang, A. Li, R. Pan, Stock trend prediction based on a new status box method and adaboost probabilistic support vector machine, Appl. Soft Comput. 49 (2016) 385–398.
[20] C.F. Tsai, Z.Y. Quan, Stock prediction by searching for similarities in candlestick charts, ACM Trans. Manag. Inf. Syst. 5 (2014).
[21] M.H. Dempster, T.W. Payne, Y. Romahi, G.W.P. Thompson, Computational learning techniques for intraday FX trading using popular technical indicators, IEEE Trans. Neural Netw. 12 (2001) 744–754.
[22] S.Y. Seidel, T.S. Rappaport, 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings, IEEE Trans. Antennas Propag. 40 (1992) 207–217.
[23] A. Bose, C.H. Foh, A practical path loss model for indoor wifi positioning enhancement, in: International Conference on Information, Communications & Signal Processing, 2007, pp. 1–5.
[24] H.Z. Moayedi, M.A. Masnadi-Shirazi, Arima model for network traffic prediction and anomaly detection, in: In 2008 International Symposium on Information Technology, Vol. 4, 2008, pp. 1–6.
[25] P.A. Dinda, D.R. O'hallaron, Host load prediction using linear models, Cluster Comput. 3 (2000) 265–280.
[26] R.C. Garcia, J. Contreras, M. Van Akkeren, J.B.C. Garcia, A GARCH forecasting model to predict day-ahead electricity prices, IEEE Trans. Power Syst. 20 (2005) 867–874.
[27] B. Chen, Y.R. Gel, N. Balakrishna, B. Abraham, Computationally efficient bootstrap prediction intervals for returns and volatilities in ARCH and GARCH processes, J. Forecast. 30 (2011) 51–71.
[28] G.P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing 50 (2003) 159–175.
[29] S. Sun, S. Wang, Y. Wei, G. Zhang, A clustering-based nonlinear ensemble approach for exchange rates forecasting, IEEE Trans. Syst. Man Cybern.: Syst. (2018).
[30] N.I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, IEEE Comput. Intell. Mag. 4 (2009) 24–38.
[31] S. Fu, Y. Li, S. Sun, H. Li, Evolutionary support vector machine for rmb exchange rate forecasting, Physica A 521 (2019) 692–704.

[32] P. Yaohao, P.H.M. Albuquerque, Non-linear interactions and exchange rate prediction: Empirical evidence using support vector regression, Appl. Math. Finance 26 (2019) 69–100.
[33] S. Sun, S. Wang, Y. Wei, A new multiscale decomposition ensemble approach for forecasting exchange rates, Econ. Model. (2018).
[34] S. Sun, Y. Wei, S. Wang, Adaboost-lstm ensemble learning for financial time series forecasting, in: International Conference on Computational Science, 2018, pp. 590–597.
[35] V. Plakandaras, T. Papadimitriou, P. Gogas, Forecasting daily and monthly exchange rates with machine learning techniques, J. Forecast. 34 (2015) 560–573.
[36] W.K. Wong, M. Manzur, B.K. Chew, How rewarding is technical analysis? Evidence from Singapore stock market, Appl. Financial Econ. 13 (2003) 543–551.
[37] T. Dietterich, Overfitting and undercomputing in machine learning, ACM Comput. Surv. 27 (1995) 326–327.
[38] D.H. Bailey, J. Borwein, M. Lopez de Prado, Q.J. Zhu, The probability of backtest overfitting, J. Comput. Financ. (2016).
[39] C.R. Harvey, Y. Liu, Evaluating trading strategies, J. Portf. Manag. 40 (5) (2014) 108–118.
[40] P. Carr, M. Lopez de Prado, Determining optimal trading rules without backtesting, 2014, Available at SSRN 2658641.
[41] F. Schorfheide, K.I. Wolpin, On the use of holdout samples for model selection, Amer. Econ. Rev. 102 (3) (2012) 477–481.
[42] D. Pradeepkumar, V. Ravi, Soft computing hybrids for forex rate prediction: A comprehensive review, Comput. Oper. Res. 99 (2018) 262–284.
[43] Dukascopy, Historical data feed, 2007-2016, https://www.dukascopy.com/swiss/english/marketwatch/historical/.
[44] T.E. Copeland, D. Galai, Information effects on the bid-ask spread, J. Finance 38 (1983) 1457–1469.
[45] M.D. Evans, R.K. Lyons, Order flow and exchange rate dynamics, J. Polit. Econ. 11 (2002) 170–180.
[46] J. Park, A market microstructure explanation for predictable variations in stock returns following large price changes, J. Financ. Quant. Anal. 30 (1995) 241–256.
[47] F. Wang, K. Yamasaki, S. Havlin, H.E. Stanley, Scaling and memory of intraday volatility return intervals in stock markets, Phys. Rev. E 73 (2006) 026117.
[48] X. Gabaix, P. Gopikrishnan, V. Plerou, H.E. Stanley, A theory of power-law distributions in financial market fluctuations, Nature 423 (2003) 267–270.