Expert Systems With Applications 257 (2024) 124948

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An intelligent option trading system based on heatmap analysis via PON/POD yields

Min-Kuan Chen [a], Dong-Yuh Yang [b], Ming-Hua Hsieh [c], Mu-En Wu [d,*]

[a] Department of Money and Banking, National Chengchi University, Taipei, Taiwan
[b] Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, Taiwan
[c] Department of Risk Management and Insurance, National Chengchi University, Taipei, Taiwan
[d] Department of Information and Finance Management, National Taipei University of Technology, Taipei, Taiwan

ARTICLE INFO

Keywords:
Quantitative trading
Options trading strategy
Winning rate forecasting
Machine learning
Heatmap analysis

ABSTRACT

In recent years, many studies have explored artificial intelligence (AI) in quantitative trading and financial prediction. Among financial products, options are highlighted for their risk management capabilities and defined trading cycles, yet they pose challenges due to their complexity. This paper proposes HAPPY (Heatmap Analysis via PON/POD Yield), an options trading system designed to predict expected winning rates (EW) by integrating actual profits, losses, and risk factors, thus enhancing traditional winning rate metrics. HAPPY employs heatmap analysis to address the risk of overfitting by smoothing isolated low EW values and incorporates machine learning (ML) models like random forest, extreme gradient boosting (XGBoost), and light gradient boosted machine (LGBM) for improved prediction accuracy. Employing TAIEX weekly options, the study evaluates EW and backtests trading performance, comparing empirical statistics and ML models. Findings indicate that ML models excel in accuracy and precision, though empirical statistics perform better in backtesting, especially as options near expiration. This research offers a robust options trading system that can be applied to other options markets or predictive models.

1. Introduction

The financial sector has witnessed a transformative surge in the application of artificial intelligence (AI) and machine learning (ML). These advancements span a multitude of domains, including but not limited to stock market forecasting, algorithmic trading, portfolio optimization, and derivatives pricing (Ahmed et al., 2022; Goodell et al., 2021). Algorithmic trading, in particular, has become a focal point due to its ability to execute trades based on detailed algorithmic analyses of temporal, price, and volume data (Ozbayoglu et al., 2020; Sahu et al., 2023). Quantitative trading, a form of algorithmic trading, stands out for its utilization of advanced mathematical models to exploit market efficiencies (An et al., 2022; Li et al., 2020; Ozbayoglu et al., 2020).

Financial market predictive model can employ traditional statistical techniques or those harnessing the power of ML (Zhou et al., 2019). Among the statistical methods, stalwarts like the autoregressive integrated moving average (ARIMA) and the generalized autoregressive conditional heteroskedasticity (GARCH) have established themselves as key tools in guiding stock market forecasts (Sirisha et al., 2022; Zhang et al., 2021). However, the foundational assumptions of linearity and stationarity in these models present inherent challenges, particularly in the volatile environment of modern financial markets. These limitations are starkly evident when compared with the dynamic, non-linear fluctuations that characterize today's stock markets (Chen et al., 2020; Lv et al., 2022).

ML stands out for its unparalleled capability to analyze nonlinear patterns within extensive datasets. These ML models, inherently proficient at navigating the non-linear and dynamic intricacies of financial asset prices, have become indispensable tools for financial time series forecasting. Their adeptness in discerning intricate patterns, augmented by technological advancements, facilitates a thorough examination of vast historical price archives. Consequently, this synergy has amplified the adoption and efficacy of advanced predictive models in the financial sector (El Majzoub et al., 2023; Henrique et al., 2019; Masini et al., 2023). For instance, Khaidem et al. (2016) developed a robust random forest algorithm to predict the direction of stock movement based on technical indicators. Nabipour et al. (2020) employed nine ML models and two DL models, including random forest, XGBoost, and recurrent neural networks (RNN), to predict stock market trends. Er and Sun (2021) used an XGBoost time-series model to forecast stock trends. Yin

* Corresponding author.
E-mail addresses: minkuanchen99@gmail.com (M.-K. Chen), yangdy@ntub.edu.tw (D.-Y. Yang), mhsieh@nccu.edu.tw (M.-H. Hsieh), mnasia1@gmail.com (M.-E. Wu).

https://doi.org/10.1016/j.eswa.2024.124948
Received 11 November 2023; Received in revised form 12 June 2024; Accepted 29 July 2024
Available online 31 July 2024
0957-4174/

et al. (2022) demonstrated that a multi-objective tree model with a light gradient boosted machine can effectively reduce the difference between model predictions and real-world returns.

Options are derivatives and represent a more complex financial instrument compared to stocks. These contracts grant the buyer the right – but not the obligation – to buy (call) or sell (put) an asset at a predetermined price upon expiration. The buyer pays a non-refundable premium, and sellers must post a margin to ensure they can fulfill the contract if exercised (Hull & Basu, 2016). Consequently, developing an options trading system, particularly one that integrates ML, poses significant challenges. Previous research on options has predominantly concentrated on pricing and hedging strategies, with ML proving advantageous in these areas (Bali et al., 2023; Ivaşcu, 2021). However, there is a relatively limited body of research specifically focused on developing quantitative trading strategies for options using ML. Options, with their myriad combinations of strike prices and positions, present formidable tools for risk management and profit optimization within trading strategies. One such option trading strategy of note is the spread strategy, renowned for its cost-saving attributes, enhanced risk management capabilities, and the capacity to assess the likelihood of maintaining positions until options reach maturity. Spread strategies are acknowledged for their quantifiable trading potential; however, they necessitate a methodology to sift through noise, and one such method is the use of expected value.

Trading based on expected values, typically derived from odds and winning rates (Kelly, 1956), poses significant forecasting challenges. While setting up odds within options trading strategies is relatively straightforward, accurately predicting winning rates is notably more complex. Wu and Chung (2018) introduced a spread strategy to calculate expected value, utilizing historical option settlement price distributions to illustrate profits and losses on the settlement day and employing the Kelly criterion to optimize fund management. While this method simplifies the analysis of profit opportunities in spread strategies, it lacks validation through backtesting to confirm its effectiveness. Further advancing this field, Wu et al. (2022) integrated ML algorithms such as k-nearest neighbors (KNN), random forest, SVM, and Naive Bayes to predict the win rates at option settlement. By incorporating ensemble learning techniques, they significantly enhanced the accuracy of win rate predictions and optimized the odds for implementing naked option trading strategies.

Traditional winning rate calculations use the number of profits divided by the number of trades. For example, in a coin-tossing game, if heads are considered a win, the winning rate is obtained by dividing the cumulative number of heads by the number of throws through repeated tossing. In other words, if the roll is heads, the numerator and the denominator are plus one; if the roll is tails, the numerator is plus zero and the denominator is plus one. Finally, the numerator and denominator are accumulated separately and divided to obtain the winning rate. However, in many financial transactions there are not only two situations: winning or losing. There may be big wins, big losses, small wins, small losses, etc. Therefore, the motivation behind this study attempts to reinterpret the winning rate using the amount of payoff. Simultaneously, the winning rate is utilized to calculate the expected value, which is then applied in the context of option trading.

Departing from traditional winning rate computations that rely solely on profit and trade counts, this paper introduces an approach known as the expected winning rate (EW), which offers a more comprehensive analysis. EW is determined by two critical components: part of the numerator (PON) and part of the denominator (POD). This method expands beyond the conventional reliance on profit and trade counts, incorporating both the actual profit magnitude and the investor's risk-reward ratio, often referred to as odds. By accumulating PON and POD over a specified timeframe and then dividing them, we derive the EW. Building on this concept, a predictive trading system for options has been developed that incorporates heatmap analysis alongside EW. The heatmap component of the methodology extends beyond mere visualization; it plays a critical role in enhancing the predictive utility of EW. Specifically, the heatmap addresses the challenge of isolated low EW values surrounded by higher values, significantly reducing the risk of overfitting in quantitative trading strategies, such as parameter plateaus within heatmaps (Wu et al., 2024). The strategic integration of heatmap analysis with predictive modeling significantly improves trading decisions by providing a nuanced interpretation of complex data patterns. Subsequently, heatmap of EW is employed to select profitable trading times and opportunities by calculating the expected value in future trading scenarios. Therefore, this paper introduces heatmap analysis via PON/POD yield (abbreviated as "HAPPY"). HAPPY is a framework explicitly designed to identify profitable opportunities within an options trading system.

Within the HAPPY, the initial step involves estimating the EW for trading spread strategies. PON, POD, and EW, previously computed, are then employed to construct matrices representing trading moments and the premium associated with spread strategies. To demonstrate the options quantitative trading methodology within the HAPPY, not only the EW obtained through empirical rules is employed but also ML techniques such as random forest, XGBoost, and LGBM are leveraged. While these techniques are classical, their integration within the novel HAPPY framework to calculate and visualize EW represents a significant advancement in the application of ML to financial trading. The rolling window method is used to train the data for both the empirical statistical and predictive models. The study integrates the strengths of these approaches by introducing the HAPPY combined with various ML models to provide a reliable framework for predicting win rates and conducting backtests of options trading strategies.

The approach is tested on the weekly Taiwan capitalization weighted stock index (TAIEX) options, and the winning probabilities and odds are determined using the spread strategy premiums. The accuracy of the expected values is evaluated using the realized payoff of the spread strategy, and the performance of the approach is analyzed using various metrics such as the win ratio, profit factor, and Sharpe ratio.

This paper makes the following contributions:

- The development of an options trading system, HAPPY, to estimate EW, which refines winning rate calculations by incorporating comprehensive risk assessments.
- The application of heatmap analysis to integrate the surrounding EW values to prevent overfitting and enhance the predictive accuracy and decision-making processes.
- Validation of the HAPPY framework's performance through empirical experiments and comparison with traditional ML models, demonstrating its utility in real-world trading scenarios.

The remainder of this paper is organized as follows. Section 2 provides description of spread strategy in options trading. Additionally, brief outline of ML models such as, random forest, XGBoost, and LGBM. Moving on to Section 3, the framework HAPPY will be explored in depth, shedding light on key indicators, the methodology for computing EW and the visualization process using heatmaps. Demonstrations of HAPPY in conjunction with ML models are also featured. Section 4 presents empirical results stemming from backtesting the spread strategy based on expected value. The accuracy of EW is assessed using a confusion matrix, and an analysis of trading strategy performance on different weekdays is conducted. Lastly, Section 5 summarizes findings, discusses their implication, and suggests potential directions for future research.

2. Preliminaries

Options are financial derivatives that offer high leverage and risk hedging. They represent a zero-sum game, as the profit result of specific strategies can be determined when options expire. Spread strategies,

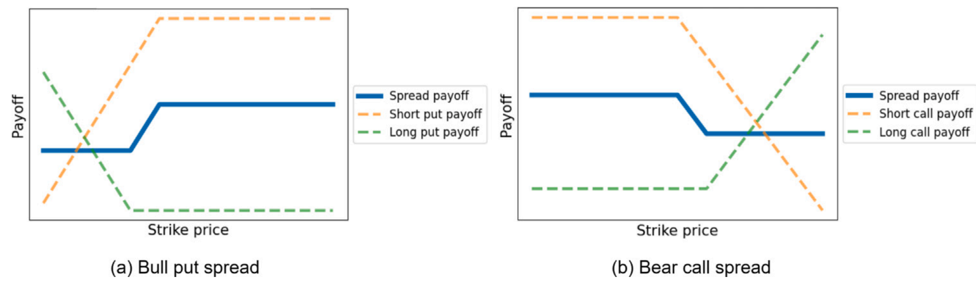

Fig. 1. (a) Bull put spread's payoff on the settlement; (b) bear call spread's payoff on the settlement.

Table 1
Components of bull put spread and bear call spread.

Spread strategy	Bull put spread	Bear call spread
Component	Long K_0 put & Short K_1 put, $K_0 < K_1 \le$ ATM	Short K_2 call & Long K_3 call, ATM $\le K_2 < K_3$
Maximum profit	$p(K_1) - p(K_0)$	$c(K_2) - c(K_3)$
Maximum loss	$K_1 - K_0 - [p(K_1) - p(K_0)]$	$K_3 - K_2 - [c(K_2) - c(K_3)]$

K_n is strike price at nth; ATM is at-the-money for options; $p(K_n)$ is put option premium with K_n; $c(K_n)$ is call option premium with K_n.

Table 2
The sample of ATM, OTM, and ITM options.

K	$c(K)$	$p(K)$	X	Call	Put
16 400	261	63	198	$ITM_{call}4$	$OTM_{put}3$
16 450	220	73	147	$ITM_{call}3$	$OTM_{put}2$
16 500	184	87	97	$ITM_{call}2$	$OTM_{put}1$
16 550	150	103	47	$ITM_{call}1$	ATM_{put}
16600	120	122	2	ATM_{call}	$ITM_{put}1$
16 650	92	143	51	$OTM_{call}1$	$ITM_{put}2$
16 700	66	168	102	$OTM_{call}2$	$ITM_{put}3$
16 750	46.5	200	153.5	$OTM_{call}3$	$ITM_{put}4$
16 800	30.5	234	203.5	$OTM_{call}4$	$ITM_{put}5$

K is the strike price; $c(K)$ is the call option premium; $p(K)$ is the put option premium.

such as vertical, butterfly, or condor spreads, can pre-lock the maximum profit and loss. The spread strategy fixes the odds and simplifies the computation of the expected value. However, the financial price is stochastic and requires modeling of nonlinear and multi-dimensional variables. ML algorithms can learn from large datasets and perform nonlinear tasks. These algorithms also improve analysis compared to the statistical approach. This section briefly introduces the spread strategy and ML models relevant to this paper.

2.1. Spread strategy

A spread strategy is a trading technique that involves the simultaneous use of either short (selling) or long (buying) options with distinct strike prices, but with the same option type (either call or put), underlying assets, and expiration dates. These strategies offer the advantage of pre-defined maximum profit and loss levels, allowing traders to assess expected values and choose optimal trading strategies. This paper focuses on two specific spread strategies: the bull put spread and the bear call spread. The components of these two spread strategies are presented in Table 1, and the corresponding payoff at settlement is illustrated in Fig. 1.

The relationship between the underlying asset price and the strike price can be classified into three types of moneyness: at-the-money (ATM), out-of-the-money (OTM), and in-the-money (ITM). ATM occurs when the strike price of an option is the same as the current market price of the underlying asset. OTM refers to an option contract that only contains extrinsic value, while ITM describes an option that has intrinsic value. For TAIEX weekly options, the strike price interval is fifty points. Therefore, ATM is biased towards the put option and call option, potentially affecting the fairness of options (both call and put). To determine the ATMs for call and put options, the steps below involve calculating the difference between call and put option.

1. Estimate the different between call and put options premiums at ith strike price and assumption $X_i = |c(K_i) - p(K_i)|$ for $i = 0, 1, \ldots$. K_i is ith strike price; $c(K_i)$ and $p(K_i)$ are the ith call and put option premiums.
2. Order X_i by ascending order to find the first index $i^* = \arg\min_i (X_i)$ and second index $i^{**} = \arg\min_{i \neq i^*}(X_i)$.
3. Define ATM for call is $ATM_{call} = \max(K_{i^*}, K_{i^{**}})$ and ATM for put is $ATM_{put} = \min(K_{i^*}, K_{i^{**}})$.
4. Define OTM for call at rth is $OTM_{call}r = ATM_{call} + 50 \times r$ for $r = 0, 1, 2, \ldots$. OTM for put at rth is $OTM_{put}r = ATM_{put} - 50 \times r$ for $r = 0, 1, 2, \ldots$.
5. Define ITM for call at rth is $ITM_{call}r = ATM_{call} - 50 \times r$ for $r = 0, 1, 2, \ldots$. ITM for put at rth is $ITM_{put}r = ATM_{put} + 50 \times r$ for $r = 0, 1, 2, \ldots$.

Given the strike price interval of fifty points in TAIEX options, OTM and ITM options are defined as having a strike price difference of plus or minus fifty points. Table 2 illustrates ATM, OTM, and ITM options, along with their respective prices on April 29, 2022, at 3:44 PM. These options correspond to the TAIEX weekly options that expired on the first Wednesday of May 2022.

2.2. Random forest

Random forest is a popular supervised ML algorithm that is considered an extension of decision trees, developed by Breiman (2001). It requires data to be artificially labeled to ensure high precision and accuracy of prediction results. The algorithm is composed of two main components, bagging (Breiman, 1996) and classification-and-regression trees (CART) (Breiman, 2017). Bagging is an extension of bootstrap aggregation for decision trees, which generates a data set by repeating random sampling with replacement from the original. CART utilizes Gini impurity (on classifications) or square errors of prediction (on regressions) to identify the best cut for each node. Random forest can be utilized for both classification and regression tasks. The average prediction across decision trees is used for regression tasks, while the majority vote class label predicted across decision trees is used for classification tasks. This algorithm has found wide application in financial research.

Significant research has been conducted in finance prediction using random forest. Basak et al. (2019) employed random forests and XGBoost to enhance long-term stock movement forecasting through decision tree ensembles. Vijh et al. (2020) utilized ANN and random forest techniques to predict stock closing prices, demonstrating their predictive capabilities. Additionally, Park et al. (2022) introduced a multitask approach that combines LSTM and RF to predict daily stock returns and their directions, outperforming baseline methods such as moving averages (MA) and the relative strength index (RSI) in predicting both returns and return directions. It performs well in both stock price prediction and trend classification.

2.3. Extreme gradient boosting (XGBoost)

XGBoost is a scalable tree-boosting system that can solve real-world tasks using minimal resources (Chen & Guestrin, 2016). It is an efficient and scalable implementation of the gradient boosting framework proposed by Friedman (2001). Unlike other methods that average independent trees, XGBoost generates a sequence of decision trees that revise the previous trees continuously by using prediction errors or residuals. This model offers high adaptability and convergence speed, up to ten times faster than comparable algorithms. XGBoost dominates the structure of predictive modeling tasks for classification and regression, allowing the modules to be parallelized and reducing training time. This technique has been successfully applied in Kaggle's competitive data science platform and is widely used in various fields, including finance.

Accurate stock price forecasting can be achieved through the application of XGBoost. In a recent study conducted by Bhatt et al. (2022), various models including LSTM, random forest, and XGBoost were employed, with XGBoost showcasing the highest level of accuracy. Additionally, Raudys and Goldstein (2022) employed LSTM and XGBoost models for forecasting both volatility and closing prices. Notably, a logarithmic transformation significantly improved the accuracy of volatility predictions. Moreover, Dezhkam and Manzuri (2023) focused on forecasting stock market trends and utilized ARIMA, XGBoost, and LSTM models. Their comprehensive analysis of the performance of these models showed that XGBoost outperformed the other models in this context.

2.4. Light gradient boosted machine (LGBM)

LGBM is a gradient boosting algorithm introduced by Ke et al. (2017) based on the gradient boosting decision tree (GBDT) technique. This algorithm is designed to overcome the limitations of many models when dealing with high feature dimensionality and large datasets, which can reduce efficiency and scalability. LGBM adopts two novel techniques, namely gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB), to overcome these limitations. GOSS determines the split point by calculating variance gain, which makes the algorithm more efficient in identifying the best split. EFB speeds up the training of the GBDT by grouping exclusive features and reducing the number of data points that the algorithm needs to scan.

Compared to conventional GBDT, LGBM is more than twenty times quicker during the training process and achieves similar accuracy. In terms of computational speed and memory consumption, LGBM significantly outperforms XGBoost. It can be applied to tabular data for regression and classification predictive modeling tasks, making it a popular choice for ML competitions.

LGBM demonstrates strong performance in stock price prediction and classification tasks. In Wang et al. (2022), it excels in predicting corporate financing risk profiles, outperforming k-nearest-neighbors, decision tree, and random forest algorithms. Sun et al. (2020) showcases LGBM's effectiveness in forecasting cryptocurrency market price trends using economic indicators, surpassing SVM and random forest models. Additionally, Zheng et al. (2022) introduces an ARIMA-LGBM hybrid model for highly accurate predictions of stock change trends.

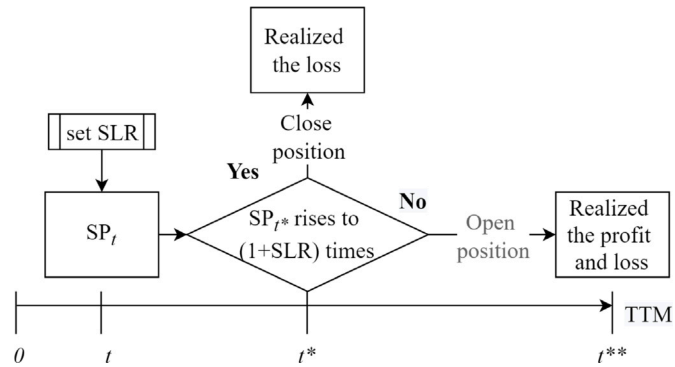

Fig. 2. Spread strategy with stop-loss mechanism.

3. HAPPY framework

This section introduces the HAPPY framework, which employs a methodology for estimating the EW of spread strategies incorporating a stop-loss mechanism. It also utilizes a heatmap for selecting profitable trading times. Initially, the section provides an overview of the spread strategy and the stop-loss mechanism. Subsequently, the PON and POD are calculated to estimate the EW. The results are presented through a heatmap that incorporates various SP and TTM values. Additionally, three popular ML algorithms, namely random forest, XGBoost, and LGBM, are employed to forecast the EW from the empirical results.

3.1. Spread strategy with stop-loss mechanism

The spread strategy represents a commonly employed options trading approach characterized by fixed profit and loss outcomes. In order to address the associated risks inherent in this strategy, a stop-loss mechanism has been introduced, as depicted in Fig. 2. This mechanism has been designed to effectively manage risk by incorporating a stop-loss ratio (SLR). Investors have the flexibility to determine their preferred SLR based on their risk tolerance. For the purposes of this paper, the SLR remains constant at 1. By implementing this mechanism, the spread strategy can receive the spread premium (SP) (refer to Eq. (1)) while simultaneously minimizing the risk of incurring losses.

$$SP = PS - PL, \tag{1}$$

where PS denotes premium of short option and PL denotes premium of long option.

In order to effectively mitigate the risk associated with the spread strategy employing the stop-loss mechanism, the strategy will be liquidated when the SP increases to (1 + SLR). Otherwise, it will be left open until the options reach their expiration. In this study, our focus centers on the weekly options contract, which possesses a time to maturity (TTM) of less than a week, thereby ensuring efficient transaction utilization. The weekly options contract is outlined as follows:

$$TTM = \{0, 1, 2, \ldots, t, \ldots, t^*, \ldots, t^{**}\}, \tag{2}$$

where t is the trading time of spread strategy with SP_t, t^* is the time of determining SP_{t^*} triggering the stop-loss standard, and t^{**} is the options expire.

3.2. PON, POD, and EW of HAPPY

The conventional approach in quantitative trading primarily assesses performance based on the count of positive realized payoffs, often overlooking risk considerations. To address this limitation, a framework named HAPPY has been introduced, as illustrated in Fig. 3. This framework is specifically applied to spread strategies within options trading and employs diverse methods for measuring EW. The

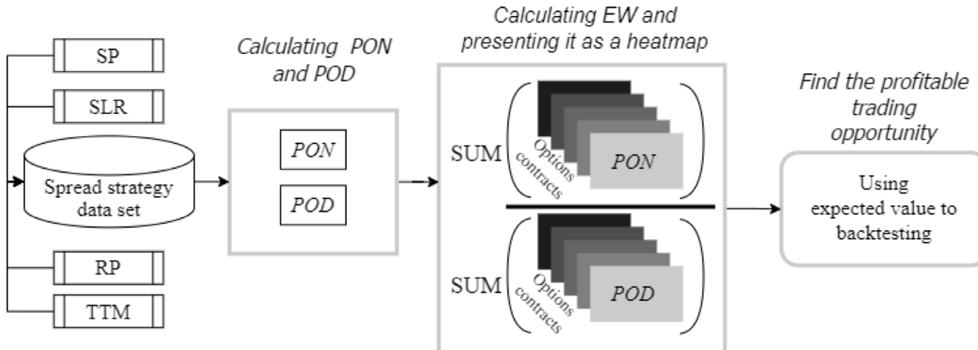

Fig. 3. Framework HAPPY: A process aimed at identifying profitable spread strategies.

dataset used in this study includes key variables such as SLR, TTM, SP, realized payoff (RP), and the type of spread strategy (bull put spread or bear call spread). PONs and PODs are computed from two distinct perspectives. Subsequently, EWs are calculated and visually represented using a heatmap.

3.2.1. Calculating PON, POD, and EW

In a coin-tossing game, where heads represent a win, the process involves tracking two key components: the numerator and the denominator. When a head is tossed, both the numerator and denominator increase by one. Conversely, when a tail comes up, the numerator remains at zero while the denominator still increases by one. The ultimate winning rate is calculated by dividing the accumulated numerator by the denominator.

However, when dealing with financial transactions, the scenario becomes more intricate due to the multitude of potential outcomes, which contrasts with the simplicity of the coin-tossing game. To address this complexity, this paper introduces two fundamental metrics: PON (analogous to the count of heads) and POD (analogous to the total number of tosses). These metrics are instrumental in computing the traditional winning rate, taking into account various factors such as maximum profit, loss, and RP. Similar to how the winning rate in a coin-tossing game emerges from repeated tossing, quantitative trading necessitates an examination of multiple past transaction records to ascertain the winning rate.

The HAPPY framework is designed as an options trading system for estimating EW, and this study demonstrates it using an option spread strategy. Traditional methods for calculating winning rates typically measure the ratio of profitable trades to total trades, but these approaches have notable shortcomings. Firstly, they neglect the magnitude of profits and losses, thus providing a limited view of a strategy's performance. Secondly, they overlook associated risks, compromising their reliability in volatile markets. To address these limitations, the proposed EW metric incorporates several refinements: EW includes both profits and losses, offering a comprehensive and realistic assessment of trading performance. The metric accounts for underlying risks, aligning the measure more closely with traders' risk tolerance. By employing the PON and POD components, the HAPPY framework facilitates a detailed breakdown of financial results, thereby enhancing the precision of the EW calculation.

The paper introduces two distinct methods (Method I and Method II) for evaluating the PON and POD, tailored to specific investment contexts. It details a step-by-step process for calculating PON and POD for each transaction, and subsequently aggregating these metrics to derive EW. These EW values are visually presented using a heatmap, laying the foundation for the HAPPY framework. Furthermore, the distinct methods for calculating PON and POD lead to the generation of two separate EW calculations. Method I and Method II are utilized to estimate the EW, addressing the diverse preferences and risk profiles of investors while maintaining a consistent foundation in risk assessment and financial outcomes analysis.

Expected value serves as a valuable tool for identifying profitable trades. The expected value calculation formula is as follows: $w \times profit + (1-w) \times loss$, where w represents the winning rate, $profit$ signifies profit points, and $loss$ denotes loss points. Throughout this paper, the assumption is consistently made that odds are 1, indicating SLR is equal to 1. To achieve a positive expected value under these conditions, w must be set to 0.5, making 0.5 the critical threshold. If w surpasses 0.5, it signifies a profitable transaction. Following this, Method I and Method II will be detailed.

Method I adopts a comprehensive allocation strategy where all realized payoffs, irrespective of whether they are profits or losses, are proportionally attributed to the PON and POD. The PON and POD using Method I are defined as Eqs. (3) and (4). When the RP equals the maximum profit or loss, it is considered a complete win; otherwise, it is a partial win. Even if the profit and loss are negative, they are still counted in PON because the maximum loss is viewed as a loss. When the profit and loss reach the break-even point, it is categorized as a no loss, no win situation. The principle applied here is that any transaction, regardless of RP, should be counted, so RP is always counted as 1.

$$PON_I := \frac{1}{2} + \frac{\text{RP}}{(1+\text{SLR}) \times \text{MP}}, \tag{3}$$

$$POD_I := \begin{cases} 1, & PON_I \in \mathbb{R}, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where PON_I and POD_I represent the PON and POD calculated using Method I, respectively. RP denotes the realized payoff of the spread strategy, while MP represents the maximum profit of the spread strategy at time t and is equal to the SP_t. SLR stands for the stop-loss ratio, and $\mathbb{R}$ denotes the set of real numbers.

For example, with SLR set to 1, MP at 20, and RP at 20, PON_I is calculated as follows: $1 = 1/2 + 20/((1+1) \times 20)$. In the same scenario with RP at 0, PON_I is calculated as $0.5 = 1/2 + 0/((1+1) \times 20)$. When RP is −20, signifying the maximum loss under the specified conditions, PON_I is $0 = 1/2 + (-20)/((1+1) \times 20)$. POD_I is consistently 1, regardless of RP's value. An example of PON_I and POD_I for various RP values is presented in Table 3.

Method II implements a selective recording strategy that records only profits in the PON, while losses are proportionately reflected in the POD. This targeted approach caters to investors primarily focused on profit maximization, emphasizing positive financial outcomes and aligning with a risk-averse investment strategy that prioritizes return on investment over a comprehensive risk assessment. The PON and POD using Method II are defined as Eqs. (5) and (6), respectively. Only positive RP values are considered in PON_{II}. If RP is 0 or negative, the transaction is categorized as a loss, resulting in a PON_{II} of 0. For POD_{II}, the larger the denominator in the winning rate, the smaller the numerator. Therefore, whenever RP equals the maximum profit or loss, 1 is counted, potentially leading to a significant accumulated POD_{II}

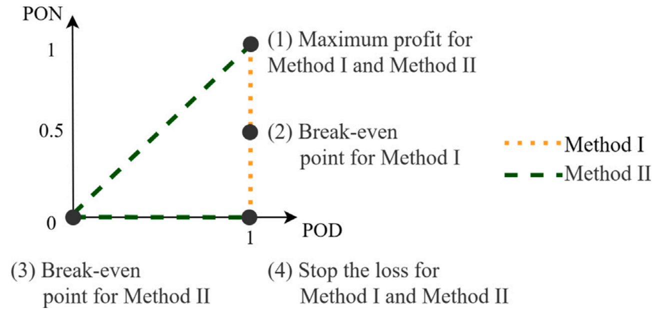

Fig. 4. Relationship between PON and POD for Method I and Method II.

value. If RP falls between the maximum profit and loss, it is considered as a partial win due to the inherent uncertainty.

$$PON_{II} := \begin{cases} \frac{\text{RP}}{\text{MP}}, & \text{RP} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

$$POD_{II} := \begin{cases} \frac{|\text{RP}|}{\text{MP}}, & |\text{RP}| \leq \text{MP}, \\ 1, & \text{otherwise}, \end{cases} \tag{6}$$

where PON_{II} and POD_{II} represent the PON and POD calculated using Method II, respectively. RP denotes the realized payoff of the spread strategy, while MP represents the maximum profit of the spread strategy at time t and is equal to the SP_t. SLR stands for the stop-loss ratio, and $\mathbb{R}$ denotes the set of real numbers.

For instance, with SLR set at 1, MP at 20, and RP at 20, PON is calculated as $1 = 20/20$. In the same conditions with RP at 0, PON is $0 = 0/20$. When RP is −20, representing the maximum loss under the specified conditions, PON is again $0 = 0/20$. Regarding POD in Method 2, with MP at 20 and RP at 20, it is calculated as $1 = 20/20$. Under the same conditions but with RP at 0, POD is $0 = 0/20$. If the conditions remain the same but RP is −20, then POD is $1 = |-20|/20$. An example of PON_{II} and POD_{II} for various RP values is presented in Table 3.

If RP is equal to MP, the EW obtained by both Method I and Method II is the same. However, if RP equals (1+SLR), the EW will be zero for both methods. The relationship between Method I and Method II is illustrated in Fig. 4. In summary, both methods consider realized payoffs, maximum profits, and risks, and aim to improve the estimation of the traditional winning rate. In other words, they do not treat trades with positive payoffs as wins without merit but estimate the winning proportion of each trade based on profit components.

Similar to the coin toss game, where the final winning rate is determined by adding and dividing the count of heads (numerator) by the total count of tosses (denominator), PON and POD are used for the calculation of EW. Leveraging historical data of option spread strategies with stop-loss conditions, multiple PON and POD values are derived based on RP. These PON and POD values are then aggregated and divided to determine EW. Subsequently, this EW is combined with the option spread strategy to calculate the expected value for future trades. Regardless of whether Method I or Method II is utilized, the formal definition of EW is as follows:

$$EW = \frac{\sum_{i=1}^{w} PON_i}{\sum_{i=1}^{w} POD_i}, \tag{7}$$

where EW denotes EW obtained through either Method I or Method II. PON_i and POD_i, representing PON and POD derived using Method I or Method II for the ith weekly options contract, are involved. The dataset's length is denoted as w. Table 3 presents a sample trading scenario for a bull put spread at various RPs. This scenario involves taking a long position with a premium of 10 for a 16,000 put option and a short position with a premium of 30 for a 16,050 put option. Both the SPt and MP are set at 20, and the SLR is set to 1. This analysis focuses specifically on single option trading. In practice, multiple

Table 3
A sample for trading a bull put spread for different RPs.

Methods	Method I			Method II		
RP	PON	POD	EW	PON	POD	EW
20	1	1	1	1	1	1
10	0.75	1	0.75	0.5	0.5	1
0	0.5	1	0.5	0	0	–[a]
−10	0.25	1	0.25	0	0.5	0
−20	0	1	0	0	1	0

[a] Symbol — signifies not-a-number due to both PON_{II} and POD_{II} being equal to zero.

option trading records are collected and analyzed, with PON and POD values being accumulated accordingly.

Method I and Method II systematically analyze actual profits and losses to calculate EW but differ significantly in how they weigh these outcomes. Such bifurcation enables the framework to accommodate a broad spectrum of investment strategies, ranging from conservative to aggressive, by adjusting the emphasis on different aspects of financial performance according to investor preferences. By offering these two calculation modalities, the experimental design facilitates a nuanced analysis of options trading strategies, enhancing the adaptability and applicability of predictive models to real-world trading environments.

The objective of this paper is to identify profitable spread strategies, specifically trading strategies with a positive expected value. EW has been derived using PON and POD, and this EW will be employed as the probability of profit under identical trading conditions in future instances. Considering the SP, TTM, and SLR of the spread strategy, the expected value as follows:

$$\mathbb{E}(\text{spread strategy}) = \text{SP} \times \text{EW} + (\text{SP} \times (-\text{SLR})) \times (1 - \text{EW}), \tag{8}$$

where $\mathbb{E}(.)$ represents the expected values of the spread strategy, EW is the expected winning rate, SP is the spread premium, and SLR is the stop-loss ratio.

For instance, if SP is 20 and SLR is 1, and the previously obtained EW from historical statistics is 0.5, then the expected value is 0. If all other conditions remain constant, and the EW becomes 0.6, then the expected value becomes 4. Likewise, if all other conditions remain the same but the EW drops to 0.4, then the expected value becomes −4.

3.2.2. Heatmap for EW and profitable trading opportunity

The objective of this paper is to identify profitable trading opportunities. It involves not only pinpointing a specific time point or spread strategy but also delving into multiple time points and various participants to calculate the expected value across different spread strategies. Weekly options are characterized by their regular weekly settlement time and trading cycle. As a result, EW can be visualized within a heatmap based on TTM or SP. While heatmaps are well-established tools for identifying profitable trading opportunities, their primary function within our framework is to visualize the EW. Our approach stems from the computation of EW using the PON and POD, which are integrated within the HAPPY framework. The heatmap not only aids in interpreting these results but also plays a pivotal role in enhancing the predictive utility of EW. Specifically, it addresses the challenge of isolated low EW values surrounded by higher values, effectively reducing the risk of overfitting in quantitative trading strategies, such as parameter plateaus within heatmaps (Wu et al., 2024).

This approach allows traders to utilize statistics from the current week when engaging in option spread trading with matching TTM and SP parameters, thereby calculating the expected value before trading the upcoming weekly option. Subsequently, the original single-point PON, POD, and EW calculations have been extended to encompass multiple TTMs and SPs. The heatmap's vertical axis represents SPs in intervals of five points, excluding the upper bounds (e.g., [0,5), [5, 10), . . .), while the horizontal axis represents TTMs.

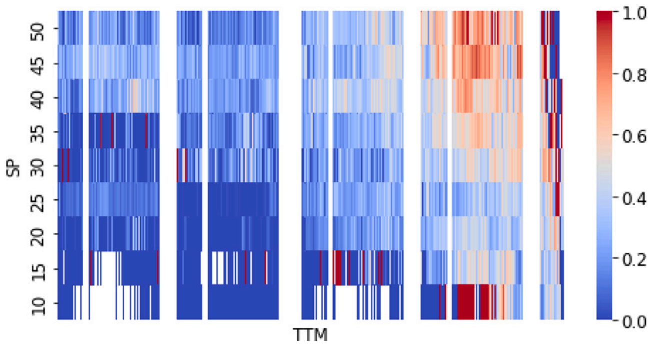

Fig. 5. Visualizing the heatmap of empirical EWs using Method I.

When representing the PON and POD of each weekly option as matrices, the PON and the POD accumulate data over a historical time period, subsequently being divided to generate the EW. This EW offers an effective means of identifying optimal trading moments and spread strategies for the weekly options contract. It forms the basis for the proposed HAPPY framework, as depicted in Algorithm 1.

Algorithm 1 HAPPY for computing PON, POD, EW, and expected value

Input: SP, TTM, w
Output: EW, E

$SP \Leftarrow$ spread premium
$TTM \Leftarrow$ time to maturity
$w \Leftarrow$ length of a sample with weekly options contracts
$EW \Leftarrow$ expected winning rate
$E \Leftarrow$ expected value
$pon \Leftarrow$ initial PON
$pod \Leftarrow$ initial POD
for i from 1 to w **do**
    $PON \Leftarrow$ calculating PON for each TTM and SP
    $POD \Leftarrow$ calculating POD for each TTM and SP
    $pon+ = PON \Leftarrow$ accumulating PON
    $pod+ = POD \Leftarrow$ accumulating POD
end for
$EW = \frac{pon}{pod}$
$E \Leftarrow$ expected value as follow Eq. (8)

The EW and heatmap are generated based on the results obtained using Method I and Method II for the rth OTM option for short positions and the type of spread (bull put spread or bear call spread). The heatmap provides insights into EW, SP, TTM, OTM, and spread types. In Fig. 5, you can observe a sample heatmap that specifically focuses on EWs, demonstrating the bear put spread at $OTM_{call}0$, with TAIEX options expiring on the fourth Wednesday of July 2022. The TTM spans from Thursday at 08:45 to the following Wednesday at 13:30. White areas on the heatmap indicate instances of either no trading activity or missing data. In the heatmap, a deep red color indicates a high EW value, approaching 1, while a deep blue color represents a low EW value, nearing 0. Given that this paper have set the SLR equal to 1, the EW must be above 0.5 to yield an expected value greater than 0. Consequently, the heatmap for EW uses 0.5 as the midpoint value, represented by a gray color. Investors can leverage this heatmap to identify specific TTM and SP combinations for weekly options contract spread strategies, potentially uncovering enhanced trading opportunities.

3.3. Prediction of EW via ML

Option values are susceptible to random fluctuations due to various factors influencing the options market, making winning rates unpredictable. While methods such as ML and neural networks can offer viable models for market sentiment, accurately forecasting winning rates remains challenging. Therefore, the recommendation is to utilize the predictive models illustrated in Fig. 6. The subsequent steps, based on HAPPY and empirical EW, aim to predict the EW and spread strategies using diverse models.

Input: The EW matrices obtained from the first weekly options contract to the wth weekly options contract using HAPPY, where w is the length of the sample with weekly options contracts.

Step 1: Define the features and target of the input data. Possible features include implied volatility, risk-free rate, and time value. The EW serves as the target.

Step 2: Train the predictive models using regression based on the target EW, which is a continuous variable ranging between 0 and 1.

Step 3: Predict EW using the input data.

Output: The EW and heatmap for the $(w+1)$th weekly options contract.

Within the HAPPY framework, three tree structure models – namely, random forest, extreme gradient boosting, and light gradient boosted machine – were employed to forecast the EW and subsequently compared with empirical EW. These models represent various forms of ensemble learning, particularly well-suited for regression problems. Random forest, which employs bagging and aggregates results for a prediction by averaging the final results across decision trees, can reduce the error of datasets with uneven and overfitting issues. Extreme gradient boosting and light gradient boosted machine are based on gradient-boosted machines that use an ensemble of weak learners to improve model performance. Extreme gradient boosting continuously revises the scalable system of the previous trees to generate new trees and is fast and accurate. On the other hand, the light gradient boosted machine is more efficient than extreme gradient boosting, as it only saves the discretized value of the feature.

To train the ML models, SP, TTM, and OTM were used as features, considering the options premium, time value, strike price, and underlying asset price. The target variable is the EW derived based on the HAPPY indicator. As the target variable is continuous, three ML models with regression were employed. The dataset was not split into training and validation data during the fitting of the ML models, as backtesting was utilized to evaluate precision and trading performance.

Finally, the three trained ML models were applied to forecast the EWs for trading spread strategies. The ML procedure in our experiments is as follows:

1. The input to the ML models is the EW matrices obtained via HAPPY for the weekly options contract.
2. The target variable is EW, and the features used are SP, TTM, and position of OTM for the short option.
3. The target and features were used to fit the ML models with regression for predicting EW, including random forest, extreme gradient boosting, and the light gradient boosted machine.
4. The EWs via ML models were represented by matrices and heatmaps.

Applying two methods (Method I and Method II) to four models (HAPPY, random forest, extreme gradient boosting, and light gradient boosted machine) resulted in eight matrices and heatmaps for every week. The confusion matrix and backtesting of trading spread strategies were applied in experiments to evaluate the precision of HAPPY and ML models. Return performance is measured by the win ratio, profit factor, and Sharpe ratio.

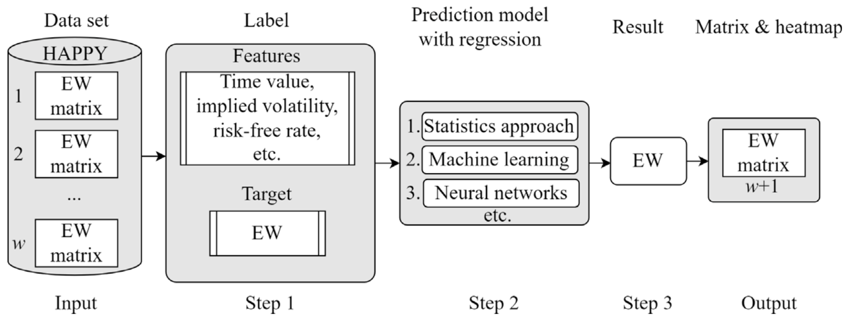

Fig. 6. Proposed predictive model.

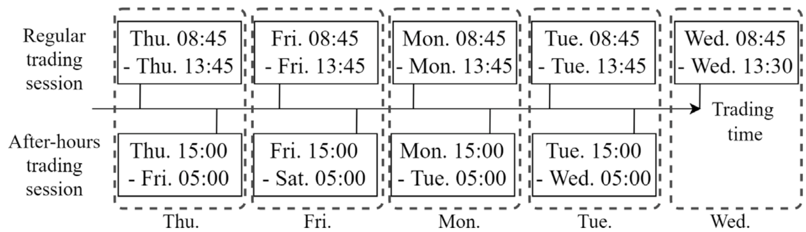

Fig. 7. Trading time of TAIEX weekly options.

4. Experiments

4.1. Datasets and protocol

The complexity of options trading and the intricacies of their contracts pose significant challenges to data collection. For this study, TAIEX weekly options data was chosen due to its accessibility and the reliability provided by the official Taiwan Futures Exchange. The dataset comprises 152 weekly TAIEX options that expired between January 2019 and December 2021. The trading time of TAIEX options is divided into regular trading sessions and after-hours trading sessions. The regular trading session is from 08:45 to 13:45, and options expire on Wednesdays at 13:30. The after-hours trading session is from 15:00 to 05:00 on the following day. Due to low liquidity and limited quoted prices at the beginning of the contract, spread strategies are not considered during this period. Hence, as depicted in Fig. 7, the trading period for each weekly TAIEX option commences at 08:45 on Thursday and concludes at 13:30 on the subsequent Wednesday. Sessions enclosed in dotted boxes are classified according to the same weekday, such as Thu. 08:45–Thu. 13:45 and Thu. 15:00–Fri. 05:00 being categorized as Thursday, and Fri. 08:45–Fri. 13:45 and Fri. 15:00–Sat. 05:00 being categorized as Friday.

The data timestamps in this paper were presented in minutes, and quoted prices were represented by open, high, low, and close prices. To minimize situations where spread strategies could not be constructed, ex-quoted prices were utilized as substitutes for missing values. The ATM_{put} and ATM_{call} had been discussed in Section 2.1. Open prices were adopted for $c(K_i)$ and $p(K_i)$. Taxes and fees were ignored in the experiments, and the low and high prices of options were used to cover the costs. Therefore, SPt was composed of a short option premium with a low price and a long option premium with a high price for conservative estimates. SPt^* comprised a short option premium with a high price and a long option premium with a low price to compensate for previous underestimations.

In this work, HAPPY and ML models were executed in Python on Google Colab, a computer programming language, and they utilized the default settings and parameters for random forest,[1] extreme gradient boosting,[2] and the light gradient boosted machine.[3]

Financial markets are subject to fluctuations driven by policies, economic factors, and other variables, making patterns at each time point unique. The rolling window approach, which has been shown to enhance the performance of stock predictive models in previous studies (Hollis et al., 2018; Matsunaga et al., 2019), was also employed in this research to analyze time series data for predicting stock prices. Consequently, this study experiments utilized a rolling window approach with weighting towards more recent time points. This method captures the most pertinent features over time and prevents overfitting of the predictive model. In this study, a window size of 48 TAIEX weekly options was used, with data in the window being weighted to emphasize the time series data. The weights were assigned based on the number of weeks before the target week, with data from the 45th to the 48th weeks receiving the highest weight (four times), and data from the 1st to the 24th weeks receiving the lowest weight (one time). The HAPPY model had used this rolling window and weighting approach to process the PON and POD matrices and compute the EW, while the three ML models had employed the same approach to obtain EW matrices from HAPPY as a sample dataset. Finally, the rolling window and weighting approach had also been used to validate the precision of EW and backtest trading (see Fig. 8).

For the short options, bull put spreads were utilized at $OTM_{put}0$, $OTM_{put}1$, and $OTM_{put}2$, as well as bear call spreads at $OTM_{call}0$, $OTM_{call}1$, and $OTM_{call}2$. Please refer to Table 4 for the symbols used to denote Method I and Method II for obtaining EW, HAPPY, and the three ML models employed in this paper.

The vertical axis in the heatmap represented SPs ranging from 10 to 50. However, SPs that were equal to or below 5 were not considered, as the spread strategy could quickly trigger a stop loss and result in misjudgment. The horizontal axis in the heatmap represented

[1] Accessed on 7 November 2023 from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

[2] Accessed on 7 November 2023 from https://xgboost.readthedocs.io/en/stable/parameter.html.

[3] Accessed on 7 November 2023 from https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html.

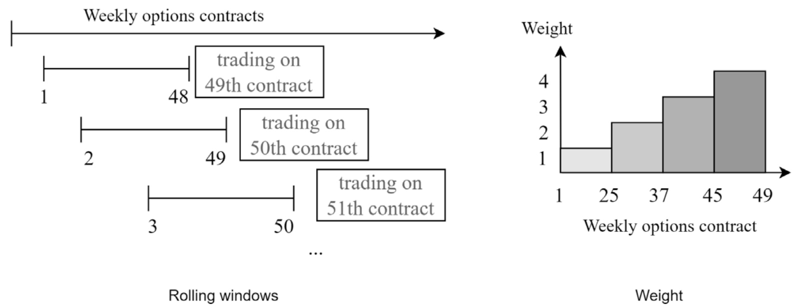

Fig. 8. Rolling window and weight analysis of time-series models.

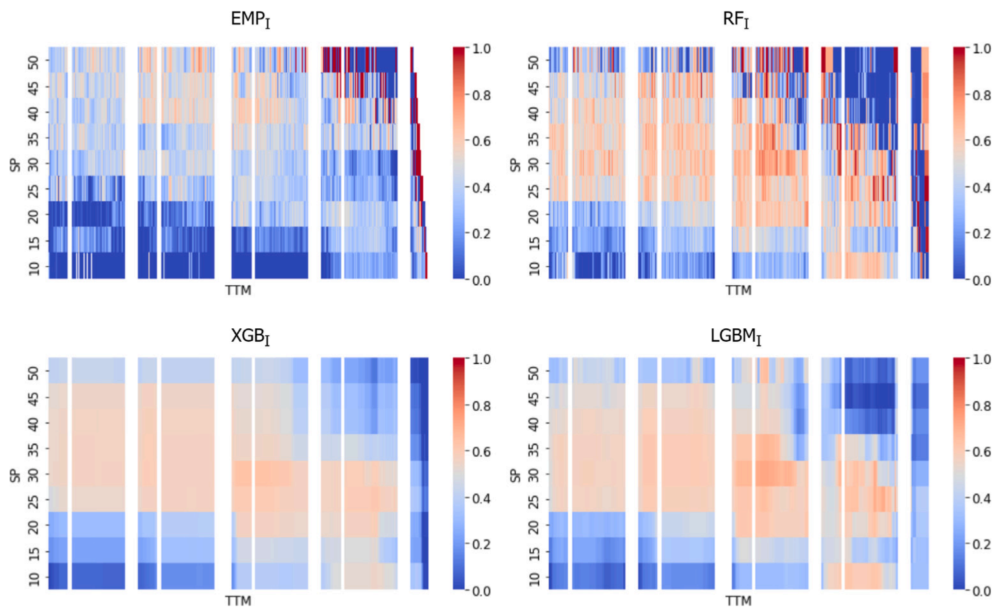

Fig. 9. EW heatmaps using Method I for bull put spread at $OTM_{call}0$.

Table 4
Description of proposed approaches.

Model_Method	Description.
EMP_I	EW by empirical statistic and Method I
RF_I	EW by random forest and Method I
XGB_I	EW by XGBoost and Method I
LGBM_I	EW by LGBM and Method I
EMP_II	EW by empirical statistic and Method II
RF_II	EW by random forest and Method II
XGB_II	EW by XGBoost and Method II
LGBM_II	EW by LGBM and Method II

timestamps (in minutes) from Thursday at 08:45 to Wednesday at 13:30, encompassing periods of no trading. Figs. 9 and 10 depicted the EW heatmaps for bull put spreads at $OTM_{put}1$ with an SLR of 1. These heatmaps were generated using Method I and Method II, respectively, and were based on TAIEX weekly options expiring on the fourth Wednesday of December 2022. Both EW heatmaps displayed similar patterns, characterized by blue areas indicating SP values below 20 at the beginning of the TAIEX weekly options and SP values above 40 at the end of the TAIEX weekly options.

In the EW heatmaps, profitable moments for spread strategies can be identified by looking for deep-red points, which represent opportunities for trading on the subsequent weekly TAIEX contract. Gray and blue points indicate that trading should be avoided. For EWs above 50%, SP falls between 15 and 35 (as shown in Fig. 11) for all methods and models. Therefore, to evaluate the precision of EW, SP between 15 and 35 was applied.

4.2. Precision of EW

When backtesting trading strategies, the primary objective was to ensure the generation of profits. However, accurately estimating the actual winning rate could be challenging. Common regression model evaluation metrics such as mean square error (MSE) or mean absolute error (MAE) were not employed in this context. Instead, the approach taken was that of a binary classification task, where the focus was on EW and stop loss as binary variables.

In this approach, a positive value for the stop-loss standard is defined as RP being equal to SP multiplied by (1+SLR), and a negative value otherwise. The evaluation utilized a confusion matrix, where the actual value represented the stop-loss standard, and the predicted value was determined by whether EW exceeded a certain threshold, denoted as z (where z could take on various levels, such as 50%, 60%, 70%, 80%, and 90%). A prediction was considered positive when EW was greater than the threshold z, and negative otherwise.

In this context, true positive (TP) corresponded to cases where EW exceeded z and the stop loss was not triggered. False positive (FP) represented cases where EW exceeded z, but the stop loss was triggered. False negatives (FN) denoted cases where EW was less than or equal to z, and the stop loss was not triggered. Lastly, true negative (TN) pertained to cases where EW was less than or equal to z, and the stop loss was triggered. This binary classification approach allowed for an

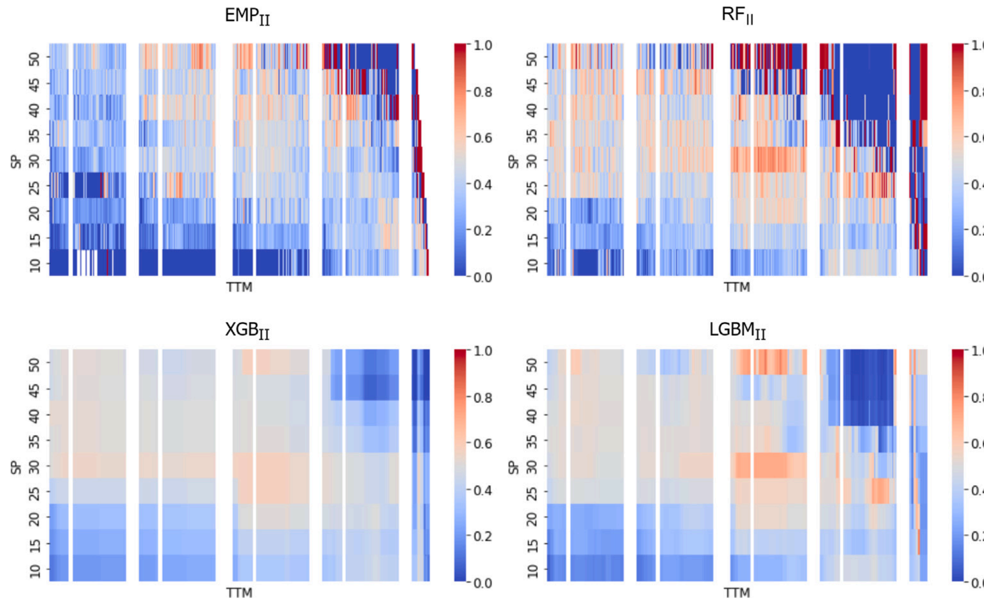

Fig. 10. EW heatmaps using Method II for bull put spread at $OTM_{call}0$.

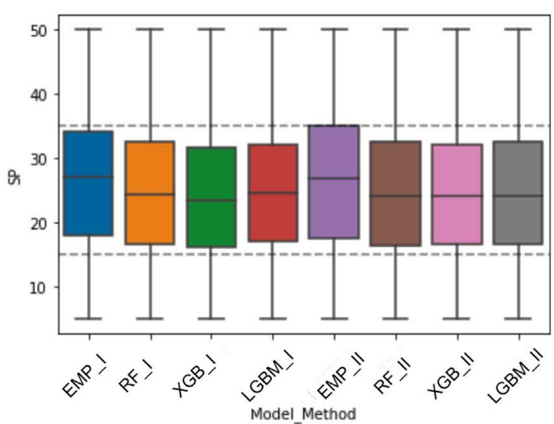

Fig. 11. SP that EWs above 50%.

assessment of trading strategy profitability while considering stop-loss criteria.

The precision is the number of times that EW was above z and did not trigger the stop loss, divided by the total number of times that EW was above z. This metric measures the proportion of times that the EW signal was accurate in predicting profitable trades. The precision is defined as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{9}$$

The accuracy measures the number of correct predictions in the overall predicted values. In other words, the overall accuracy of EW and realized payoff can be evaluated. The accuracy is defined as:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{10}$$

The recall measures the number of actual results that are not stopped by the loss and have an EW above z, divided by the total number of actual results. It captures the proportion of positive results (i.e., profitable spread strategies) that are correctly identified by the model. The recall is defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{11}$$

In this study, a confusion matrix was employed, which was based on a stop-loss mechanism, to categorize the EW. This allowed for the computation of precision, accuracy, and recall for the proposed approach. The outcomes of the confusion matrix are presented in Table 5.

The primary concern in evaluating the proposed methods was whether the EW and realized payoff could match, i.e., whether the trades generated under the given EW condition could avoid hitting the stop-loss standard. Therefore, precision was the most important metric. However, XGB_II and LGBM_II had many periods with an accuracy of 0% due to TP equaling 0. This meant that no trades were found that did not hit the stop-loss, indicating that every trade hit the stop-loss. Among the models, XGB_I (with an EW of 70% on Fridays) showed the highest precision at 72.64%. Precision was considered meaningful only when it exceeded 50%, which indicated that at least half of the trades could avoid hitting the stop-loss through EW. As shown in Fig. 12, none of the methods had precision exceeding 50% on Thursdays (the front-end weekly expiry date of TAIEX options). However, methods gradually exceeded 50% after Friday, especially on Tuesdays and Wednesdays. It was speculated that this was because as the weekly expiry date of TAIEX options approached (which was on Wednesday), the uncertainty was greatly reduced compared to the beginning of trading, making it easier to evaluate whether stop-loss was reached through EW.

Regarding recall, the measurement assessed how many times the EW condition was met among all trades that did not hit the stop-loss. The highest recall was observed in XGB_I (79.76%) (EW of 50% on Tuesdays). However, this metric was not our primary concern since our main focus was on increasing the ratio of trades that did not hit the stop-loss based on the given EW condition, as reflected in precision. In most cases, the recall reached 0 at an EW of 90%, indicating too many FN where trades did not meet the EW condition but did not hit the stop-loss standard. As previously discussed in precision, many methods had a high precision due to a significant number of TP, which was our primary concern in enhancing overall performance.

From the overall performance of the predictions, ML models were generally better than empirical statistic. However, ML models had too many instances of FN when not triggering the stop-loss and not meeting the EW conditions. In contrast, empirical statisti's predictions were not outstanding but more robust and consistent, unlike ML models that had large differences in performance under different conditions or weekdays. At the same time, it was found that precision will be higher as the option expiration date approaches (i.e., on Tuesdays and

Table 5
Precision, accuracy, and recall via empirical statistic and ML models.

Weekday	EW	Indicators	EMP_I	RF_I	XGB_I	LGBM_I	EMP_II	RF_II	XGB_II	LGBM_II
Thu.	>50%	Precision (%)	36.33	30.62	31.64	30.43	**37.12**	29.93	31.49	28.97
		Accuracy (%)	62.08	53.32	56.13	54.09	**63.36**	51.59	53.25	50.53
		Recall (%)	23.80	35.52	31.34	33.20	20.14	37.62	**38.55**	37.03
	>60%	Precision (%)	38.23	30.72	**47.71**	34.40	35.12	28.84	27.78	27.46
		Accuracy (%)	66.25	61.75	**67.77**	66.18	65.86	60.24	64.80	63.88
		Recall (%)	8.02	14.97	2.06	5.66	7.30	**16.07**	5.89	7.48
	>70%	Precision (%)	**38.05**	30.96	0.00	0.00	32.48	28.33	0.00	0.00
		Accuracy (%)	67.65	66.82	**67.83**	**67.83**	67.37	66.18	**67.83**	**67.83**
		Recall (%)	0.93	2.47	0.00	0.00	1.39	**3.34**	0.00	0.00
	>80%	Precision (%)	28.00	30.39	0.00	0.00	**31.58**	28.75	0.00	0.00
		Accuracy (%)	67.82	67.71	**67.83**	**67.83**	67.78	67.67	**67.83**	**67.83**
		Recall (%)	0.05	0.22	0.00	0.00	0.20	**0.32**	0.00	0.00
	>90%	Precision (%)	21.43	14.29	0.00	0.00	**27.84**	15.89	0.00	0.00
		Accuracy (%)	**67.83**	67.79	**67.83**	**67.83**	67.82	67.81	**67.83**	**67.83**
		Recall (%)	0.01	0.01	0.00	0.00	**0.06**	0.01	0.00	0.00
Fri.	>50%	Precision (%)	**42.66**	39.94	40.09	41.13	40.80	39.88	40.27	40.08
		Accuracy (%)	60.63	57.18	57.92	58.58	**60.68**	56.55	56.57	56.21
		Recall (%)	31.28	44.81	41.88	43.58	26.76	48.48	50.80	**51.69**
	>60%	Precision (%)	**44.37**	40.67	44.17	38.86	40.93	41.13	33.02	37.84
		Accuracy (%)	63.36	60.81	**64.22**	60.38	63.21	60.62	59.76	59.72
		Recall (%)	12.07	26.56	9.92	23.81	10.98	**30.02**	14.97	24.15
	>70%	Precision (%)	43.10	41.49	**72.64**	48.02	39.67	41.09	48.97	52.50
		Accuracy (%)	64.25	64.06	65.20	65.09	64.45	63.50	65.11	**65.29**
		Recall (%)	1.81	7.05	0.29	1.60	2.54	**10.77**	1.94	4.68
	>80%	Precision (%)	38.59	**44.27**	0.00	0.00	35.28	39.66	0.00	0.00
		Accuracy (%)	64.43	65.03	**65.13**	**65.13**	64.82	64.89	**65.13**	**65.13**
		Recall (%)	0.11	0.47	0.00	0.00	0.33	**1.30**	0.00	0.00
	>90%	Precision (%)	36.73	**47.83**	0.00	0.00	35.50	37.95	0.00	0.00
		Accuracy (%)	64.45	65.07	**65.13**	**65.13**	64.89	65.12	**65.13**	**65.13**
		Recall (%)	0.02	0.01	0.00	0.00	**0.07**	0.04	0.00	0.00
Mon.	>50%	Precision (%)	47.57	46.29	46.63	46.40	**49.04**	45.25	45.14	45.59
		Accuracy (%)	53.08	50.87	51.09	50.95	**53.83**	49.57	49.21	49.91
		Recall (%)	32.68	50.45	52.83	51.31	39.99	52.24	**54.83**	52.92
	>60%	Precision (%)	**50.75**	46.58	45.73	47.48	49.28	45.54	44.58	45.32
		Accuracy (%)	**54.80**	52.00	51.56	52.63	54.31	51.12	51.10	50.88
		Recall (%)	15.58	38.12	35.44	**40.15**	17.62	38.70	31.28	39.21
	>70%	Precision (%)	55.83	47.79	**62.44**	52.71	50.38	46.62	52.88	47.74
		Accuracy (%)	55.11	53.65	**55.47**	55.36	54.58	53.18	55.00	53.86
		Recall (%)	5.47	**21.28**	4.98	16.90	5.33	20.96	8.91	16.37
	>80%	Precision (%)	**60.88**	49.12	0.00	27.31	52.88	47.71	0.00	53.41
		Accuracy (%)	**54.82**	54.41	54.56	54.51	54.61	54.20	54.56	54.62
		Recall (%)	1.39	**8.28**	0.00	0.08	1.36	8.21	0.00	0.99
	>90%	Precision (%)	**62.09**	50.62	0.00	0.00	55.87	47.51	0.00	0.00
		Accuracy (%)	**54.65**	54.58	54.56	54.57	54.58	54.41	54.56	54.56
		Recall (%)	0.31	2.97	0.00	0.00	0.38	**3.29**	0.00	0.00
Tue.	>50%	Precision (%)	55.76	51.95	50.18	51.62	**58.62**	51.32	49.00	50.01
		Accuracy (%)	52.62	51.49	47.93	50.01	**55.65**	50.41	45.97	47.66
		Recall (%)	45.28	72.30	**79.76**	73.27	51.70	69.32	77.53	70.86
	>60%	Precision (%)	60.65	52.74	48.89	51.94	**61.67**	51.62	47.63	50.58
		Accuracy (%)	52.97	51.95	46.39	49.69	**54.81**	50.55	44.86	48.27
		Recall (%)	28.54	60.93	51.76	53.26	35.92	**61.12**	53.19	51.65
	>70%	Precision (%)	**66.11**	52.86	52.83	53.72	64.24	51.99	46.86	51.29
		Accuracy (%)	52.19	51.36	48.67	50.09	**52.78**	50.49	46.08	48.59
		Recall (%)	17.52	47.93	18.57	34.25	21.92	**48.59**	21.96	35.59
	>80%	Precision (%)	**68.73**	52.79	33.13	51.37	65.73	51.70	47.40	50.38
		Accuracy (%)	50.44	**50.63**	46.80	48.05	50.46	49.83	47.25	47.81
		Recall (%)	9.50	**35.38**	1.54	16.35	11.02	37.16	6.57	20.50
	>90%	Precision (%)	**66.21**	49.97	26.47	39.41	65.14	50.23	54.10	44.78
		Accuracy (%)	48.63	48.69	47.30	46.97	**48.80**	48.71	47.82	47.02
		Recall (%)	3.52	25.04	0.35	2.24	4.54	**29.39**	2.45	5.14

(*continued on next page*)

Table 5 (*continued*).

Weekday	EW	Indicators	EMP_I	RF_I	XGB_I	LGBM_I	EMP_II	RF_II	XGB_II	LGBM_II
	>50%	Precision (%)	53.35	55.53	47.20	50.45	56.34	**57.12**	46.40	55.09
		Accuracy (%)	51.55	54.21	46.81	48.82	**54.97**	54.44	47.19	53.06
		Recall (%)	34.49	69.62	45.16	28.61	48.89	**70.52**	40.32	33.02
	>60%	Precision (%)	52.12	56.50	48.42	53.75	55.94	**57.12**	43.68	55.94
		Accuracy (%)	50.13	**54.54**	48.59	49.61	52.78	54.21	46.59	52.23
		Recall (%)	18.68	63.00	28.17	14.65	31.38	**68.85**	25.76	20.97
	>70%	Precision (%)	49.13	56.13	41.11	51.28	54.15	**57.44**	42.29	56.97
Wed.		Accuracy (%)	49.19	53.55	46.96	48.73	50.69	**54.19**	48.08	51.52
		Recall (%)	9.19	57.78	11.68	6.66	16.50	**66.12**	12.24	12.38
	>80%	Precision (%)	47.97	57.56	40.66	32.73	49.67	57.55	42.86	**57.83**
		Accuracy (%)	49.14	**54.26**	47.96	47.88	49.36	53.88	49.17	51.16
		Recall (%)	5.06	53.16	6.72	1.25	7.15	**63.11**	6.80	8.54
	>90%	Precision (%)	47.36	54.54	35.73	28.48	46.46	**57.64**	38.16	53.86
		Accuracy (%)	49.16	50.58	48.85	48.06	49.10	**53.70**	49.85	50.33
		Recall (%)	3.42	**42.27**	1.64	0.64	3.92	61.30	1.44	4.42

Bold indicate the best values in the row. Round off all values to the 2nd decimal place.

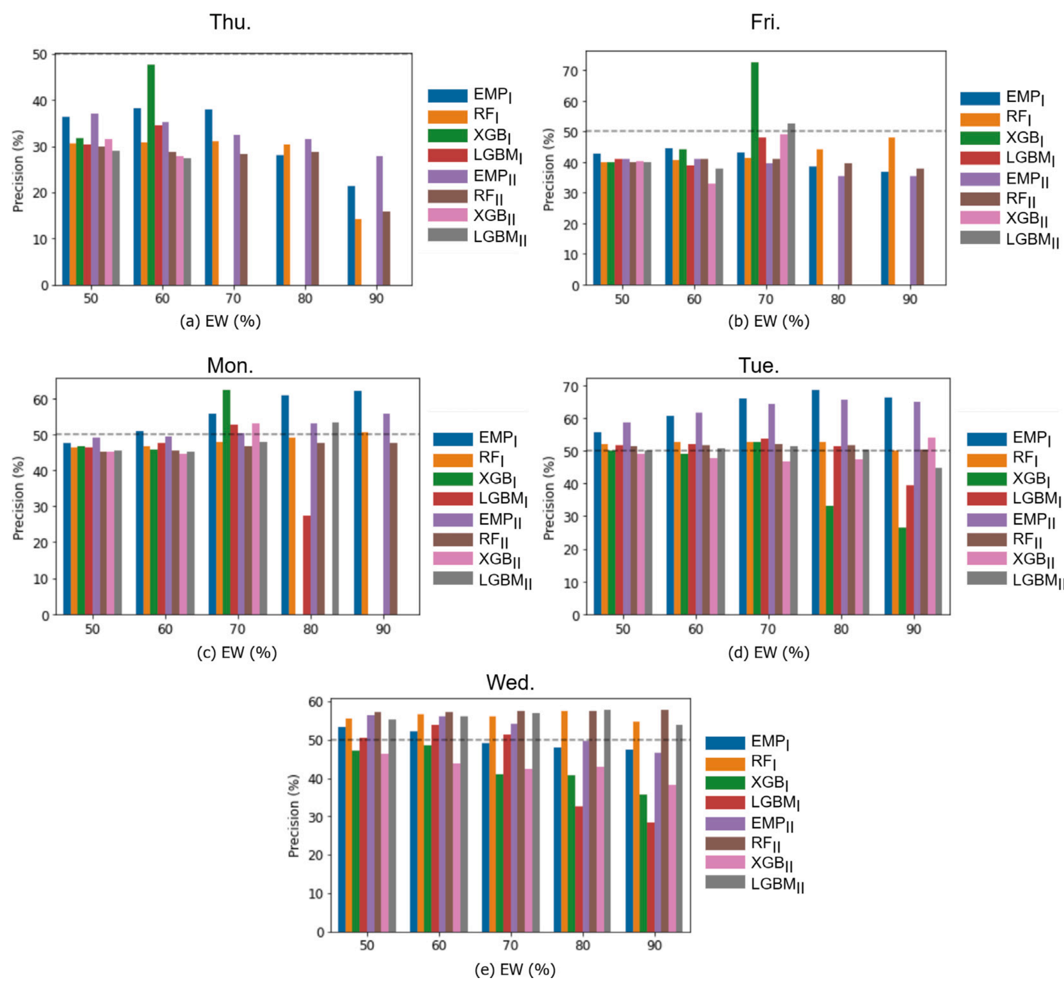

Fig. 12. Precision of EWs for each weekday.

Wednesdays), which was similar to the conclusion that futures and spot prices will eventually converge. When TTMs were shorter, uncertainty was reduced, leading to a notable enhancement in accuracy under the specified EW conditions, all without triggering the stop-loss criterion.

4.3. Backtesting of spread strategies

The EW and heatmap were utilized to identify profitable spread strategies and favorable moments for the upcoming TAIEX weekly options. To showcase the application of EW, Eq. (8) was used to execute the trading spread strategy in the subsequent experiment.

When the majority of EW values exceeded 50%, the SP fell within the range of 15 to 35, as demonstrated in Fig. 11. Most of the positive expected values were below 15, as illustrated in Fig. 13. Consequently, the expected values were categorized into three levels, and the following conditions were employed to perform a backtest on the trading spread strategies derived from the proposed approach.

1. 15 < SP < 35

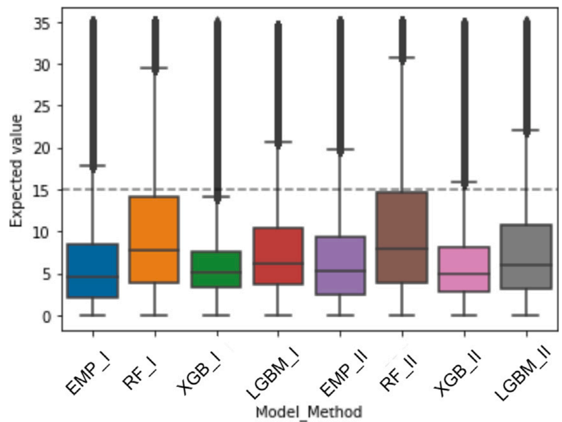

Fig. 13. Expected values where SPs fall within 50.

2. expected values $> v$, $v = 0, 5, 10$

The number of trades increased as the closing of TAIEX weekly options approached, but there were fewer trades on the expiry date, as shown in Fig. 14. This suggested that the time value and uncertainty were beneficial for options with many trading opportunities on Tuesdays. It is also noteworthy that trades on Wednesdays were significantly less frequent than on other weekdays due to the outflow of time value and low volatility for the TAIEX weekly options.

Multiple indicators were employed to evaluate the return and payoff of the proposed approach, including the win ratio, profit factor (Solberg, 1957), and Sharpe ratio (Sharpe, 1966). To prevent any confusion between the winning rate and EW, the winning rate was renamed as the *win ratio*, representing the probability of winning in traditional quantitative trading and serving as a valuable metric for analyzing trading performance. The win ratio can be expressed as:

$$\mathrm{WR} = \frac{\mathrm{NPRP}}{\mathrm{NT}}, \tag{12}$$

where WR represents the win ratio, NPRP represents the number of positive realized payoffs, and NT represents the number of trades.

The profit factor is defined as the gross profit divided by the absolute value of the gross loss. It indicates the amount of profit earned per unit of loss. A profit factor greater than 1 indicates that the trades have a positive net profit. The profit factor can be expressed as:

$$\text{Profit factor} = \frac{\mathrm{GP}}{|\mathrm{GL}|}, \tag{13}$$

where PF represents the profit factor, GP represents the gross profit, and GL represents the gross loss.

The Sharpe ratio is an indicator that assesses the benefit that can be earned for a given unit of volatility. It is calculated by dividing the mean return by the standard deviation of the return. In this study, the risk-free rate was not considered. The Sharpe ratio can be expressed as:

$$\mathrm{SR} = \frac{\mathrm{mean}(R)}{\mathrm{std.}(R)}, \tag{14}$$

were SR represents the Sharpe ratio; R denotes the historical sequence of return rates; mean(R) represents the mean R; and std.(R) represents the standard deviation of R.

The win ratios for each trading strategy of expected value were shown in Table 6. The trading strategies with expected values >0 had a win ratio ranging from 28.68% to 57.94%. For each trading week, the best performing method was using EMP_II. The trading strategies with expected values >5 had a win ratio ranging from 28.16% to 61.38%. For each trading week, the best performing method was using EMP_I. The trading strategies with expected values >10 showed that LGBM_I performs the best on Thursdays with a win ratio of 80.56%. It was important to note that XGB_I's win ratio of 0% on Thursdays was due to the fact that there were no instances where the expected value was greater than 10. However, this trading strategy was relatively unstable compared to the other two strategies (expected value >0 and expected value >5). Overall, an increasing trend in win ratios was observed as Wednesday, the day of TAIEX weekly options expiry, approaches.

Table 6
Win ratio (%) with different expected values.

Expected value	Model_Method	Weekday Thu.	Fri.	Mon.	Tue.	Wed.
>0	EMP_I	36.25	**42.52**	47.33	55.18	52.61
	RF_I	30.15	39.91	46.12	51.40	54.76
	XGB_I	31.50	40.07	46.51	49.60	45.96
	LGBM_I	30.10	41.11	46.28	51.06	48.79
	EMP_II	**36.84**	40.53	**48.58**	**57.94**	55.60
	RF_II	29.36	39.85	45.03	50.76	**55.94**
	XGB_II	31.21	40.24	45.02	48.43	45.58
	LGBM_II	28.63	40.06	45.43	49.49	53.56
>5	EMP_I	**38.22**	**44.06**	**51.20**	60.65	51.44
	RF_I	31.35	41.04	47.06	52.54	55.62
	XGB_I	28.16	41.29	47.02	48.58	47.27
	LGBM_I	31.33	40.07	48.21	51.96	50.55
	EMP_II	35.70	40.77	48.89	**61.38**	**55.98**
	RF_II	29.84	41.84	45.99	51.34	55.97
	XGB_II	33.14	38.13	47.41	46.96	43.32
	LGBM_II	32.58	39.32	45.95	50.53	53.99
>10	EMP_I	39.45	44.07	54.80	**65.33**	49.73
	RF_I	32.82	41.68	49.68	52.76	55.55
	XGB_I	0.00	45.76	**56.29**	50.25	44.32
	LGBM_I	**80.56**	48.43	52.67	53.44	46.33
	EMP_II	33.30	38.93	50.57	63.46	53.66
	RF_II	30.90	42.03	48.55	51.58	**55.95**
	XGB_II	18.01	**46.12**	49.54	44.18	44.11
	LGBM_II	37.01	40.11	48.86	49.83	54.00

Bold indicates the best values in the weekday and expected values block. Round off all values to the 2nd decimal place.

Table 7 displayed the profit factors for each trading strategy. A profit factor greater than 1 indicated an overall profitable trading strategy. For trading strategies with expected values greater than 0, profit factors ranged from 0.42 to 1.38, with EMP_I and EMP_II being the more favorable methods each week. However, for Tuesday and Wednesday, there were more instances of profit factors greater than 1, indicating poorer performance on other weekdays. For trading strategies with expected values greater than 5, profit factors ranged from 0.39 to 1.57, with EMP_I and EMP_II being the more favorable methods each week. However, from Thursday to Monday, the profit factors for each method were mostly below 1, indicating relatively poor performance for these three weekdays. For trading strategies with expected values greater than 10, profit factors ranged from 0 to 4.14, with LGBM_I being the most favorable method each week. The profit factor of 0 for XGB_I was because no trading instances met the expected value criteria of greater than 10. EMP_I and EMP_II had promising performance in terms of profit factor in general, suggesting that these trading strategies could have been more stable and reliable than the others.

The Sharpe ratios for each trading strategy were presented in Table 8. Among the trading strategies with expected values greater than 0, the lowest Sharpe ratio was −0.881%, while the highest was 1.551% for LGBM_II on Wednesday, and EMP_II performed best on multiple weekdays with positive Sharpe ratios. Among the trading strategies with expected values greater than 5, the lowest Sharpe ratio was −0.351%, and the highest was 3.765%. LGBM_I had the highest Sharpe ratio of 3.765% on Wednesday, while LGBM_II had positive Sharpe ratios on five weekdays, and EMP_I had positive Sharpe ratios on four weekdays with overall better performance. Among the trading strategies with expected values greater than 10, XGB_I had the best performance on Wednesday with a Sharpe ratio of 7.753%, while

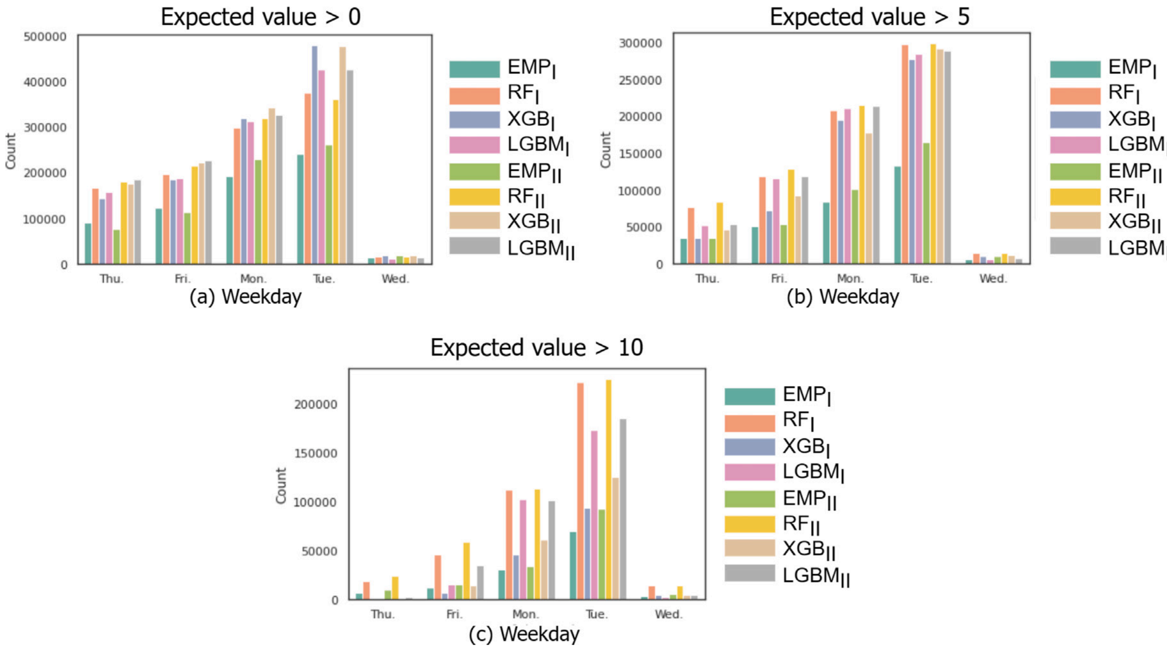

Fig. 14. Number of trades with different expected values.

Table 7
Profit factor with different expected values.

Expected value	Model_Method	Weekday Thu.	Fri.	Mon.	Tue.	Wed.
>0	EMP_I	**0.57**	**0.74**	0.90	1.23	1.11
	RF_I	0.45	0.68	0.88	1.04	1.20
	XGB_I	0.47	0.68	0.89	0.96	0.81
	LGBM_I	0.45	0.71	0.89	1.03	0.96
	EMP_II	0.59	0.68	**0.94**	**1.38**	**1.25**
	RF_II	0.44	0.68	0.84	1.02	**1.25**
	XGB_II	0.48	0.70	0.84	0.92	0.79
	LGBM_II	0.42	0.69	0.85	0.96	1.18
>5	EMP_I	**0.62**	**0.78**	**1.03**	1.51	1.06
	RF_I	0.47	0.70	0.91	1.08	1.23
	XGB_I	0.39	0.70	0.89	0.90	0.87
	LGBM_I	0.45	0.67	0.94	1.05	0.98
	EMP_II	0.56	0.68	0.94	**1.57**	**1.27**
	RF_II	0.44	0.72	0.87	1.03	1.25
	XGB_II	0.50	0.63	0.89	0.84	0.72
	LGBM_II	0.49	0.66	0.86	0.98	1.19
>10	EMP_I	0.65	0.79	1.15	**1.85**	0.98
	RF_I	0.50	0.71	0.97	1.07	1.21
	XGB_I	0.00	0.83	**1.24**	0.95	0.76
	LGBM_I	**4.14**	**0.93**	1.08	1.08	0.93
	EMP_II	0.50	0.63	0.98	1.71	1.15
	RF_II	0.45	0.72	0.93	1.03	**1.25**
	XGB_II	0.23	0.85	0.95	0.76	0.75
	LGBM_II	0.59	0.65	0.92	0.95	1.17

Bold indicates the best values in the weekday and expected values block. Round off all values to the 2nd decimal place.

Table 8
Sharpe ratio (%) with different expected values.

Expected value	Model_Method	Weekday Thu.	Fri.	Mon.	Tue.	Wed.
>0	EMP_I	0.617	0.049	−0.154	0.063	1.516
	RF_I	0.057	−0.064	0.247	0.048	−0.881
	XGB_I	**0.731**	0.357	0.001	−0.131	0.489
	LGBM_I	0.364	−0.153	0.063	−0.025	0.840
	EMP_II	−0.014	**0.591**	**0.271**	**0.213**	0.401
	RF_II	−0.201	0.068	−0.129	0.157	0.865
	XGB_II	0.377	0.296	0.224	−0.070	−0.619
	LGBM_II	−0.237	0.073	−0.168	−0.006	**1.551**
>5	EMP_I	**1.447**	−0.059	**0.481**	0.453	1.079
	RF_I	0.624	0.043	−0.192	0.364	−0.642
	XGB_I	−0.113	−0.277	−0.092	0.220	−0.056
	LGBM_I	−0.351	0.174	−0.133	−0.176	**3.765**
	EMP_II	1.277	−0.340	−0.241	**0.853**	−0.480
	RF_II	0.143	**0.649**	0.015	0.122	−0.609
	XGB_II	0.779	0.210	−0.193	0.002	1.357
	LGBM_II	0.082	0.211	0.206	0.271	0.822
>10	EMP_I	−0.098	0.308	0.167	0.436	−0.076
	RF_I	0.277	−0.162	−0.078	0.033	0.432
	XGB_I	0.000	−0.884	**0.740**	0.650	**7.753**
	LGBM_I	−8.956	0.870	0.405	0.347	1.657
	EMP_II	−0.881	**1.166**	−0.664	−0.325	−1.841
	RF_II	0.706	0.407	0.404	−0.218	−1.487
	XGB_II	**4.811**	−1.018	0.721	**0.735**	−1.569
	LGBM_II	1.728	0.085	0.306	−0.239	2.059

Bold indicates the best values in the weekday and expected values block. Round off all values to the 2nd decimal place.

LGBM_I had the worst performance on Thursday with a Sharpe ratio of −8.956%. None of the methods had consistent results under this trading strategy condition. It was possible that the opportunities for this trading strategy with expected values greater than 10 were limited, which led to less prominent Sharpe ratio performance. Overall, even when trading based on positive expected values, there were negative Sharpe ratios. No significant increase in Sharpe ratios was observed as the option expiration date approached. When using ML models for trading, they showed better performance than using HAPPY, which performed relatively weakly.

This section employed expected value for conducting backtesting and evaluating various performance indicators for different expected values and methods. It was observed that as the option expiration date approaches, performance improves in terms of trade frequency, win ratio, and profit factor. However, the Sharpe ratio's performance was not satisfactory, possibly due to significant RP fluctuations each time, resulting in a large standard deviation of returns. The ML models also exhibited a higher trade frequency than HAPPY, which may be attributed to an overestimation of expected value during calculation, implying that the EW estimate was too high.

When comparing backtesting performance, the EW of empirical statistics outperformed the ML models because its expected value was more conservative, reducing the chance of encountering losses during backtesting. However, the ML models' target was based on EW of empirical statistics, so if there was a desire to reduce the overestimation of EW or expected value in the future, it was recommended to adjust the EW of empirical statistics or the parameters in the ML models. Finally, EW was used and integrated into two algorithms, Method I and Method II, considering both trading strategy losses and weekly option trading cycles. This produced an EW heatmap each week, allowing investors to calculate profitable expected values based on different trading times and spread strategies (i.e., TTM and SP).

5. Conclusions

This paper proposes an options trading system named HAPPY, which leverages a combination of empirical statistics and ML models to estimate EW and identify profitable spread strategies. The traditional reliance on simple profit counts and total trade numbers is replaced with the PON and POD, integrating actual payoffs and associated risks into the EW calculation. The heatmap not only aids in interpreting these results but also significantly enhances the predictive utility of EW by addressing isolated low EW values amidst higher values, thus mitigating the risk of overfitting in quantitative trading strategies.

The HAPPY framework integrates classical machine learning models such as random forest, XGBoost, and LGBM to predict EW. While these models are well-established, their incorporation within the innovative HAPPY framework, which accounts for actual profits, losses, and risk factors, represents a significant advancement in applying AI to financial trading. The study, focused on TAIEX weekly options, revealed that empirical statistics outperformed the three ML models in backtesting, particularly on Tuesdays and Wednesdays as the expiry of TAIEX weekly options approached, demonstrating higher accuracy and precision. An increase in win ratios and profit factors was observed as the expiry date neared. However, the ML models tended to overestimate EWs and expected values, leading to more trades but reduced overall trading performance compared to empirical statistics.

In conclusion, this study presents a reliable and effective framework for forecasting winning rates and conducting backtests of trading strategies. Given that the fundamental trading principles and contractual nature of options are consistent across most financial markets, the HAPPY system is potentially applicable to various options products in future research. Although the study focuses on TAIEX, future work will include evaluations on multiple datasets from different markets to validate the robustness and versatility of the HAPPY framework.

The HAPPY system also holds potential for extension to other prediction models, such as reinforcement learning (RL) and long short-term memory (LSTM). While HAPPY shows promise, certain limitations must be acknowledged. The experiment is based on historical data from TAIEX weekly options and requires further verification across other markets or different trading environments. Financial markets are dynamic, and despite using a rolling window in the experiment, trading strategies must adapt to varying market conditions. Factors such as transaction costs, liquidity risk, and slippage may affect the actual profitability of the strategy. Future research could explore integrating fundamental and technical analysis, econometric analyses, and high-frequency data to further enhance the accuracy and precision of the proposed system.

CRediT authorship contribution statement

Min-Kuan Chen: Conceptualization, Methodology, Writing – original draft, Software. **Dong-Yuh Yang:** Validation, Formal analysis, Writing – review & editing, Data curation. **Ming-Hua Hsieh:** Software, Validation, Writing – review & editing, Supervision. **Mu-En Wu:** Conceptualization, Methodology, Data curation, Writing – review & editing, Resources, Project administration.

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.

Data availability

Data will be made available on request.

References

Ahmed, S., Alshater, M. M., El Ammari, A., & Hammami, H. (2022). Artificial intelligence and machine learning in finance: A bibliometric review. *Research in International Business and Finance*, *61*, Article 101646.

An, B., Sun, S., & Wang, R. (2022). Deep reinforcement learning for quantitative trading: Challenges and opportunities. *IEEE Intelligent Systems*, *37*(2), 23–26.

Bali, T. G., Beckmeyer, H., Moerke, M., & Weigert, F. (2023). Option return predictability with machine learning and big data. *The Review of Financial Studies*, *36*(9), 3548–3602.

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552–567.

Bhatt, R., Thakur, G., & Sapra, L. (2022). Forecasting the direction of stock trends using machine learning and Twitter. *Mathematical Statistician and Engineering Applications*, *71*(4), 2729–2738.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Chen, Q., Zhang, W., & Lou, Y. (2020). Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. *IEEE Access*, *8*, 117365–117376.

Dezhkam, A., & Manzuri, M. T. (2023). Forecasting stock market for an efficient portfolio by combining XGBoost and Hilbert–Huang transform. *Engineering Applications of Artificial Intelligence*, *118*, Article 105626.

El Majzoub, A., Rabhi, F. A., & Hussain, W. (2023). Evaluating interpretable machine learning predictions for cryptocurrencies. *Intelligent Systems in Accounting, Finance and Management*.

Er, X., & Sun, Y. (2021). Visualization analysis of stock data and intelligent time series stock price prediction based on extreme gradient boosting. In *2021 international conference on machine learning and intelligent systems engineering* (pp. 272–279). IEEE.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232.

Goodell, J. W., Kumar, S., Lim, W. M., & Pattnaik, D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, *32*, Article 100577.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226–251.

Hollis, T., Viscardi, A., & Yi, S. E. (2018). A comparison of LSTMs and attention mechanisms for forecasting financial time series. arXiv preprint arXiv:1812.07699.

Hull, J. C., & Basu, S. (2016). *Options, futures, and other derivatives*. Pearson Education India.

Ivaşcu, C.-F. (2021). Option pricing using machine learning. *Expert Systems with Applications*, *163*, 113799.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*.

Kelly, J. L. (1956). A new interpretation of information rate. *The Bell System Technical Journal*, *35*(4), 917–926.

Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. arXiv preprint arXiv:1605.00003.

Li, L., Wang, J., & Li, X. (2020). Efficiency analysis of machine learning intelligent investment based on K-means algorithm. *IEEE Access*, *8*, 147463–147470.

Lv, P., Wu, Q., Xu, J., & Shu, Y. (2022). Stock index prediction based on time series decomposition and hybrid model. *Entropy*, *24*(2), 146.

Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2023). Machine learning advances for time series forecasting. *Journal of Economic Surveys*, *37*(1), 76–111.

Matsunaga, D., Suzumura, T., & Takahashi, T. (2019). Exploring graph neural networks for stock market predictions with rolling window analysis. arXiv preprint arXiv:1909.10660.

Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., & Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, *8*, 150199–150212.

Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, *93*, Article 106384.

Park, H. J., Kim, Y., & Kim, H. Y. (2022). Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing*, *114*, Article 108106.

Raudys, A., & Goldstein, E. (2022). Forecasting detrended volatility risk and financial price series using LSTM neural networks and XGBoost regressor. *Journal of Risk and Financial Management*, *15*(12), 602.

Sahu, S. K., Mokhade, A., & Bokde, N. D. (2023). An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: Recent progress and challenges. *Applied Sciences*, *13*(3), 1956.

Sharpe, W. F. (1966). Mutual fund performance. *The Journal of Business*, *39*(1), 119–138.

Sirisha, U. M., Belavagi, M. C., & Attigeri, G. (2022). Profit prediction using Arima, Sarima and LSTM models in time series forecasting: A Comparison. *IEEE Access*, *10*, 124715–124727.

Solberg, H. J. (1957). The profit factor in fire insurance rates. *Journal of Insurance*, 24–33.

Sun, X., Liu, M., & Sima, Z. (2020). A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters*, *32*, Article 101084.

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, *167*, 599–606.

Wang, D.-n., Li, L., & Zhao, D. (2022). Corporate finance risk prediction based on LightGBM. *Information Sciences*, *602*, 259–268.

Wu, M.-E., & Chung, W.-H. (2018). A novel approach of option portfolio construction using the Kelly criterion. *IEEE Access*, *6*, 53044–53052.

Wu, J. M.-T., Lin, W.-Y., Huang, K.-W., & Wu, M.-E. (2024). On the design of searching algorithm for parameter plateau in quantitative trading strategies using particle swarm optimization. *Knowledge-Based Systems*, *293*, 111630.

Wu, M.-E., Syu, J.-H., & Chen, C.-M. (2022). Kelly-based options trading strategies on settlement date via supervised learning algorithms. *Computational Economics*, *59*(4), 1627–1644.

Yin, T., Du, X., Zhang, W., Zhao, Y., Han, B., & Yan, J. (2022). Real-trading-oriented price prediction with explainable multi-objective optimization in quantitative trading. *IEEE Access*.

Zhang, W., Gong, X., Wang, C., & Ye, X. (2021). Predicting stock market volatility based on textual sentiment: A nonlinear analysis. *Journal of Forecasting*, *40*(8), 1479–1500.

Zheng, X., Cai, J., & Zhang, G. (2022). Stock trend prediction based on ARIMA-LightGBM hybrid model. In *2022 3rd information communication technologies conference* (pp. 227–231). IEEE.

Zhou, F., Zhou, H.-m., Yang, Z., & Yang, L. (2019). EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*, *115*, 136–151.