KeAi
CHINESE ROOTS
GLOBAL IMPACT

Available online at www.sciencedirect.com

ScienceDirect

The Journal of Finance and Data Science 8 (2022) 162–179

http://www.keaipublishing.com/en/journals/jfds/

Trading the FX volatility risk premium with machine learning and alternative data

Thomas Dierckx [a,b,*], Jesse Davis [b], Wim Schoutens [a]

[a] Department of Statistics and Risk, KU Leuven, Celestijnenlaan 200B, Leuven, 3001, Belgium
[b] Department of Computer Science, KU Leuven, Celestijnenlaan 200A, Leuven, 3001, Belgium

Received 31 January 2022; revised 7 July 2022; accepted 11 July 2022
Available online 15 July 2022

Abstract

In this study, we show how both machine learning and alternative data can be successfully leveraged to improve and develop trading strategies. Starting from a trading strategy that harvests the EUR/USD volatility risk premium by selling one-week straddles every weekday, we present a machine learning approach to more skillfully time new trades and thus prevent unfavorable ones. To this end, we build probability-calibrated Random Forests on various predictors, extracted from both traditional market data and financial news, to predict the closing Sharpe ratio of short one-week delta-hedged straddles. We then demonstrate how the output of these calibrated machine learning models can be used to engineer intuitive new trading strategies. Ultimately, we show that our proposed strategies outperform the original strategy on risk-based performance measures. Moreover, the features that we derived from financial news articles significantly improve the performance of the approach.

Keywords: Trading strategy; Machine learning; Financial news; Alternative data; Volatility risk premium

1. Introduction

The price of an option contract is determined by, among other things, the expected risk, or volatility, of the underlying asset for the duration of the contract. It is extremely difficult to predict future volatility. In fact, it is well known that the market tends to overestimate future volatility when trading option contracts.[1] In other words, the volatility implied by option prices, known as implied volatility, often overestimates the historical volatility. The difference between implied and historical volatility is better known as the volatility risk premium, which in turn is a popular target for many trading strategies. Indeed, market participants attempt to isolate and trade this premium through a range of complex derivative strategies.

Most existing studies that investigate trading the volatility premium are situated in stock markets and report underwhelming results (e.g.[2–4]). The presence of the premium fluctuates over time, making it hard to trade profitably.

* Corresponding author. Department of Statistics and Risk, KU Leuven, Celestijnenlaan 200B, Leuven, 3001, Belgium.
E-mail address: thomas.dierckx@kuleuven.be (T. Dierckx).
Peer review under responsibility of China Science Publishing & Media Ltd.

https://doi.org/10.1016/j.jfds.2022.07.001

However, recent work by Société Générale suggests the existence of a steady volatility premium on the EUR/USD currency pair.[5] They propose a trading strategy where a new delta-hedged at-the-money straddle with seven days to maturity is systematically sold on a daily basis and show that their approach was profitable throughout the last decade. Naturally, their strategy periodically suffers from disappointing results and on average one out of three trades ends up incurring a loss. Our goal is to improve their approach by reducing the number of loss-making trades. Specifically, we investigate whether machine-learned models trained on both market and alternative data can identify on which days the strategy is likely to make money, and hence should be employed.

The combination of machine learning and alternative data is a promising approach within computational finance. In recent years, the field of finance has seen an explosion of interest in more exotic sources of information to serve alongside traditional market data. Academic literature suggests that machine learning can be used to extract valuable insights from sources such as social media (e.g.[6,7], news (e.g.[8,9]), and earning reports (e.g.[10,11]) for a variety of different applications. A key distinguishing characteristic of these alternative data sources is that they are typically textual in nature. This is in contrast with traditional market data which is numerical and readily used with modern statistical methods. The ability to extract and quantify information residing in text is therefore an essential problem to solve.

The contribution of this study is two-fold. First, we demonstrate that Random Forests trained on historical market conditions can predict the closing Sharpe ratio of short one-week delta-hedged straddles on EUR/USD. In addition, we propose a number of features that can be derived from financial news and show that using them results in improved performance compared to solely using market-based features. Second, we show how predictions from probability-calibrated Random Forests can be used in developing new and improved trading strategies. Empirically, our strategies outperform the original one out-of-sample based on risk-based performance measures.

The following sections are structured as follows: Section 2 first details necessary background information on methods used in our study, Section 3 describes our data acquisition and preparation steps, Section 4 outlines the methodology used to study our research objectives, Section 5 then presents the results of our experiments together with a discussion, after which Section 6 offers a conclusion on the performed work.

2. Preliminaries

This section covers necessary background knowledge on methods used throughout this study. Section 2.1 explains how European currency options are priced, Section 2.2 outlines the dynamically delta-hedged straddle, Section 2.3 briefly describes topic modelling with Latent Dirichlet Allocation, and lastly Section 2.4 defines Random Forests.

2.1. Pricing european-style currency options

The Garman-Kohlhagen option pricing model[12] is a well-known method to valuate European-style currency options. The model adapts the famous Black-Scholes model[13] in order to cope with the presence of two risk-free interest rates. More formally, the domestic currency value of a call C and put P European option contract can be calculated as:

$$C = S_0 e^{-r_f T}\mathcal{N}(d_1) - Ke^{-r_d T}\mathcal{N}(d_2) \tag{1}$$

$$P = Ke^{-r_d T}\mathcal{N}(-d_2) - S_0 e^{-r_f T}\mathcal{N}(-d_1) \tag{2}$$

with:

$$d_1 = \frac{\ln(S_0/K) + (r_d - r_f + \sigma^2/2)T}{\sigma\sqrt{T}} \tag{3}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{4}$$

and where:

S_0	the current spot rate
K	the strike price
$\mathcal{N}(x)$	the cumulative normal distribution function
r_d	the domestic risk-free interest rate
r_f	the foreign risk-free interest rate
T	the time to maturity
σ	the volatility of the spot rate

There are three different partial derivatives in Equation (1), known as Greeks, that are of interest for this study: delta (Δ), gamma (G), and theta (T). They can be computed as:

$$Delta \frac{\delta C_t}{\delta S_t} = e^{-r_f T}\mathcal{N}(d_1) \tag{5}$$

$$Gamma \frac{\delta^2 C_t}{\delta S_t^2} = \frac{\mathcal{N}^{'}(d_1)e^{-r_f T}}{S_t\sigma\sqrt{T}} \tag{6}$$

$$Theta \frac{\delta C_t}{\delta T} = \frac{-S_t\mathcal{N}^{'}(d_1)\sigma e^{-r_f T}}{2\sqrt{T}} + r_f S_t \mathcal{N}(d_1)e^{-r_f T} - r_d K e^{-r_d T}\mathcal{N}(d_2) \tag{7}$$

Intuitively, delta quantifies the rate of change between the option price and a \$1 move in the underlying asset, gamma quantifies the rate of change in the delta of an option for a \$1 move in the underlying asset, and theta quantifies the rate of change in the option price for a one-day change in time to the option expiration date. Theta is also known as time decay.

2.2. Harvesting the volatility risk premium with straddles

The dynamically delta-hedged straddle is a frequently used approach in the domain of volatility trading that attempts to capture the volatility risk premium on an asset. The strategy consists of selling, or buying, both a put and call option with (a) equal duration and (b) strike prices that lie as close as possible to the current spot price. On inception, the position is quasi delta-neutral and mainly exposed to the volatility of the underlying asset. However, when the price of the underlying asset diverges, the position becomes increasingly exposed to price direction. This exposure can be minimized by regularly re-hedging to maintain delta-neutrality. More concretely, the position is systematically re-hedged by buying, or selling, a certain amount of equity in the underlying asset specified by:

$$Hedge(t) = -\big(\Delta_{call}(t) + \Delta_{put}(t)\big) * Contract\ Unit \tag{8}$$

where $\Delta_{call}(t)$ and $\Delta_{put}(t)$ respectively denote the delta of the call and put option at time t, and where *Contract Unit* denotes the total number of underlying units an option contract controls.

Given an array of simplifications, such as the sticky strike rule, and a short re-hedge period, the P&L of a short delta-hedged straddle at time t can be approximated by[14]:

$$P\&L(t) = \Theta(t-1)\Delta t - \frac{1}{2}\Gamma(t-1)\Delta S^2 \tag{9}$$

where Θ and Γ represent the net theta and gamma of the straddle, and where Δt and ΔS represent the change in time and underlying price.

Equation (9) makes clear that the P&L of the strategy does not depend on the price direction of the underlying asset. Instead, it depends on the size of time decay, determined by the implied volatility when the position was opened and time left to maturity, and the magnitude of price moves in between hedging, which represents realized volatility. Put

another way, when implied volatility is expected to overstate realized volatility, it makes sense to short a delta-hedged straddle as the time decay will outsize the loss associated with price swings, and vice versa.

2.3. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA)[15] is a natural language processing technique and generative probabilistic method that automatically discovers hidden topics in a text corpus. The topic structures correspond to distributions over words, and they are inferred in a way so that they maximize the likelihood of generating the documents present in the corpus. The main idea behind the method is that each document can be represented by a random distribution over K topics. Consequently, each document can then be produced by repeatedly sampling a topic structure from which a word is then drawn. Ultimately, the technique can transform a text document into a vector of length K denoting a mixture over the hidden topics. More specific, each document D in a corpus can be represented as:

$$D = (P(k_1), P(k_2), P(k_3), \ldots, P(k_K)) \tag{10}$$

where $P(k_i)$ indicates the probability of hidden topic k_i being present in document D.

The hyperparameter K, which controls how many latent topics are retrieved, is not known beforehand. Choosing the right value for K is not straightforward, as there is typically no prior knowledge about how many hidden topic structures are present within a corpus.

2.4. Random Forests

Random Forests[16] are a popular machine learning approach for learning a predictive model. They consist of multiple different decision (or regression) trees (e.g., built with CART[17] on random subsets of samples and predictors) whose predictions are combined into one final prediction. The combination is typically done by taking the mode (or average) of all outputs. For example, the final prediction for a regression problem can be obtained by:

$$\widehat{y}_i = \frac{1}{M}\sum_{m=1}^{M} f_m(x_i) \tag{11}$$

where f is a function in the set of all possible decision trees, and M is the total number of trees in the ensemble.

The advantages of Random Forests include that they are fast to build, are not affected by feature scaling, are robust to irrelevant predictors, and are robust to noisy data.[18] Moreover, their method of constructing an ensemble model reduces the risk of overfitting on the training data.

3. Data preparation

In this section, we describe the data used in our study. We first outline how we acquired and prepared both market and news data in Section 3.1. Then, we show how we historically simulated the systematic trading strategy proposed by Société Générale in Section 3.2. Lastly, Section 3.3 details how we extracted predictors from both historical market and news data to use in our machine learning setup.

3.1. Market and news data

We acquired data ranging from January 1st, 2000 to December 31st, 2020 from two different sources:

1. **Bloomberg** where we obtained the end-of-day EUR/USD price, at-the-money implied volatility, and the USD and EUR interest rates.
2. **Refinitiv** where we obtained all English news articles that cover macro-economic and foreign exchange news supplied by Reuters News.

Note that end-of-day denotes 5:00 pm ET, and that we only considered data on weekdays. We did not remove holidays that fall on weekdays.

The news articles from Refinitiv typically consist of a headline, body, and a variety of metadata such as publication date, article type, and subject codes that relate to the topics found in the news item. We used this metadata to select a subset of the corpus to include in our study. First, because we are investigating a strategy on EUR/USD, we only selected articles containing subject codes connected to the United States (G:6 J, M:Y) or Europe (G:3, G:B4, G:AL, M:I, M:K). Second, not all news categories are equally relevant for our currency pair. We therefore filtered our corpus by only selecting articles containing at least one relevant hand-picked subject code such as currency and money markets, central banks, monetary and fiscal policy, and general macro-economic news (A:2, A:8, A:9, A:*n*, E:5, E:9, E:A, E:B, E:C, E:4 S, M:E9). Note that Refinitiv changed its internal subject encoding system during our time period from *n*2000 to RCS Qcode. We therefore converted their legacy *n*2000 codes into the current RCS Qcodes prior to any processing. All codes mentioned are RCS Qcodes and they are further explained in A. Additionally, we removed subject codes from article metadata that did not represent topic codes, such as Reuters Instrument Codes, and Refinitiv PermIDs.

The news format of the selected articles is not homogeneous, and they typically follow one of three distinct formats: articles with tabular data and minimal text, articles made up of short disjointed summaries of multiple news stories, and normal news articles with coherent text. We found that the format of an article can typically be inferred by looking at the Refinitiv headline tag. If present, this tag is located in front of the news headline and denotes a Refinitiv-specific keyword. Using headline tags, we further refined our news collection by only retaining normal news articles by selecting articles with (i) no tag or (ii) the *FOREX* tag present. This selection was based on trial-and-error as we found no documentation about the headline tagging system. For each article, we also removed all text between angle and square brackets because this text typically contains advertisements for other Refinitiv products, and we converted the publication date timezone from UTC to ET.

3.2. Baseline strategy simulation

We simulated the trading strategy proposed by Société Générale[5] for the period that spans January 1st, 2000 to December 31st, 2020. Every weekday, a new over-the-counter at-the-money EUR/USD straddle is sold with a duration of five trading days and a notional value of $1 MM. Each position is delta-hedged on a daily basis and held until maturity. The historical option prices were computed with the Garman-Kohlhagen model using the historical at-the-money EUR/USD implied volatility that corresponds with the option maturity.

Several assumptions were made for the simulation. First, we assumed that all trading actions were done at 5:00 pm ET without liquidity issues, and that holidays were absent. Second, we assumed that option orders were filled at the historical price computed by the Garman-Kohlhagen model, and that orders in the underlying asset were filled at mid-price. Third and last, we assumed that transaction and slippage costs were negligible.

The P&L of each short delta-hedged straddle was logged on every trading day at 5:00 pm ET, and was then used to measure its closing Sharpe ratio.[19] We adapted the ratio to measure the average profit earned for the risk taken, as defining a return on a short option position is not straightforward. More formally, the closing Sharpe ratio of each short delta-hedged straddle was computed as:

$$\textit{Sharpe Ratio}^{*} = \frac{\mathbb{E}[P\&L]}{\sigma_{P\&L}} \tag{12}$$

where $\mathbb{E}[P\&L]$ and $\sigma_{P\&L}$ represent the expected value and standard deviation of the daily P&L. A higher Sharpe ratio is associated with a better trade.

3.3. Feature engineering

The predictors we used in our study come from four different origins. First, Section 3.3.1 documents how we extracted features from market information, time information, and trade information. Second, Section 3.3.2 details how we developed features from financial news. Third and last, Section 3.3.3 describes how we derived additional features for each original one by quantifying their inherent temporal information. Note that all features are constructed on a daily level where each day starts at 5:01 pm on day t_{i-1} and ends at 5:00 pm on day t_i.

3.3.1. Market, time, and trade features

We collected a total of ten different features from market, time and trade information. First, we chose features that are directly related to the performance of the trading strategy. To this end, we obtained daily end-of-day values for EUR/USD returns, 30-day implied volatility, and 30-day rolling realized volatility. In addition, we computed the daily implied volatility rank with a look back period of 252 trading days, and the daily difference between implied and rolling realized volatility. Second, as recent work has shown that the strategy may be affected by seasonality,[5] we computed daily values for current day of the week, month, and financial quarter. These values were denoted by ordinal numbers. Third and last, we reason that trade performance might be short-term autocorrelated. Consequently, we derived daily trade features that quantify the recent performance of the trading strategy. We computed the hit rate, where a hit indicates a trade with positive profit, and average Sharpe ratio (Section 3.2) on a rolling basis with a look back period of 22 trading days. Note that it takes exactly five trading days for a trade to conclude. This means that for a hit rate on day t_i, we take the trades placed on days t_{i-5} to t_{i-5-22}. Table 1 summarizes the obtained features per origin.

3.3.2. News features

We applied three different techniques to transform text into numerical features: counting items, topic modelling via Latent Dirichlet Allocation, and lexicon-based processing.

The first family of features was obtained by simply counting publications, subject codes, and alerts. Given a subset of news articles X, we derived three types of news concentration indicators we expect might affect trade performance. First, $Attention(X)$ measures the daily media attention on news segment X compared to all news publications. Second, $Dispersion(X)$ measures the daily average number of subjects per article for a given news segment X. Third and last, $Urgency(X)$ measures the daily proportion of news articles flagged as alerts compared to all news publications for news segment X. These indicators are defined more formally as:

$$Attention(X) = \frac{Publications(X)}{Publications(all)} \tag{13}$$

$$Dispersion(X) = \frac{Codes(X)}{Publications(X)} \tag{14}$$

$$Urgency(X) = \frac{Alerts(X)}{Publications(X)} \tag{15}$$

where $Publications(X)$, $Codes(X)$, and $Alerts(X)$ represent counting functions for respectively the daily number of publications, unique RCS Qcodes, and alerts in a given news subset X. Alerts are identified by the metadata field *urgency* having a value of three, or the more recently introduced field *messageType* having a value of one. We computed these indicators for six news subsets that respectively cover articles on Europe, the US, either Europe or the US, the ECB, the Federal Reserve, and either the ECB or the Federal Reserve. This yielded a total of 18 features. In addition, we modelled the difference in news concentration between relevant European and United States news by introducing three polarity measures:

$$Attention\ Polarity(X, Y) = Attention(X) - Attention(Y) \tag{16}$$

Table 1
This table presents the obtained features for three different information sources. Each feature is of daily granularity where each entry represents the end-of-day value. Each market feature is based on EUR/USD.

Market	Time	Trade
Returns	Day Of Week	Rolling Hit Rate
Implied Volatility (IV)	Month	Rolling Sharpe Ratio
Realized Volatility (RV)	Quarter	
Implied Volatility Rank		
IV-RV Difference		

$$Dispersion\ Polarity(X, Y) = Dispersion(X) - Dispersion(Y) \tag{17}$$

$$Urgency\ Polarity(X, Y) = Urgency(X) - Urgency(Y) \tag{18}$$

We computed these polarities between the news subset covering Europe and the subset covering the US, and the news subset covering the ECB and the subset covering the Federal Reserve. This resulted in six additional features. In total, we obtained 24 distinct counting features.

The second family of features was obtained by using Latent Dirichlet Allocation on bag-of-words representations of article bodies from the whole news corpus. We processed the body of each article using the Python package Spacy.[20] Every article was tokenized, after which each word was converted to lowercase and lemmatized. We removed common stop words, e-mail addresses, URLs, punctuation, and named entities corresponding to dates, units of time, monetary values, measurements, percentages, ordinals, and other cardinal numbers. We trained three different topic models for $k \in \{10, 20, 40\}$ using the Python package Gensim.[21] The hyperparameter configuration is listed in B, together with an illustration of latent topics found in our corpus. Ultimately, given a training corpus, we transformed it into a temporally ordered feature matrix of $N \times K$ where each row represents an article as a mixture over K latent themes. In turn, we averaged the feature vectors of articles on the same day, transforming the $N \times K$ matrix into a matrix of dimension $T \times K$ where each row now denotes the mixture over K latent themes on day t. Lastly, we used this feature matrix to derive an entropy indicator that measures how focused news media are reporting over K different latent topics. More concretely, we applied the Shannon entropy formula[22] on each feature vector yielding a new temporally ordered feature where each entry denotes the news entropy on a given day. Note that we train topic models only using articles in the training set. They are then applied to the out-of-sample articles in the test set.

The third and last family of features was obtained through the Python text processing package Textblob.[23] This package uses a simple lexicon-based approach that extracts a sentiment and subjectivity measure for each article. More concretely, it computes the average sentiment and subjectivity over each adjective using a lexicon of words and their hand-tagged scores. Sentiment is a real value in the interval $[-1, 1]$ and quantifies the sentiment present. Subjectivity is a real value in the interval $[0, 1]$ and measures the amount of personal opinion present. Note that Textblob uses pattern matching to account for negated adjectives, and that the employed lexicon is SentiWordNet[24] which is not specifically tailored towards language used in financial articles (in contrast to e.g.25). This process results in a feature matrix of dimension $N \times 2$ where each row denotes the sentiment and subjectivity found in a given article. We averaged feature vectors from articles published on the same day, which in turn resulted in the feature matrix of dimension $T \times 2$ where each row now denotes the average sentiment and subjectivity found in articles published on day t. Table 2 summarizes the different types of news features extracted per technique.

3.3.3. Encoding temporal information

Most machine learning models, such as Random Forests, assume independence among data points. This means they will not be able to leverage temporal information unless it is explicitly encoded in a feature. For example, our model will not be able to see that a feature value is higher today than yesterday without designing a feature to capture this pattern. Therefore, we construct four additional predictors per feature to capture the following temporal trends: the daily difference (or first-order difference), the exponential moving average, the difference between the daily value and this moving average, and the standard deviation. The moving average and standard deviation were computed on a rolling basis using a window of 22 trading days. Table 3 summarizes the number of original and total features after this feature engineering process per feature source.

Table 2
This table presents the derived news features for each technique. Each feature is of daily granularity where each entry represents the end-of-day value.

Counting[a]	LDA	Lexicon
Attention	K Latent Topics	Sentiment
Dispersion	Topic Entropy	Subjectivity
Urgency		

[a] Note: includes the original and polarity function.

Table 3
This table shows the number of features per information source before and after time-encoding features are added. Note that variable K represents the LDA hyperparameter that controls the number of hidden topics modelled.

	Market	Time	Trade	News	All
Original	5	3	2	$27 + K$	$37 + K$
Total	25	15	10	$135 + 5K$	$185 + 5K$

4. Methodology

Our study breaks down into two research objectives:

1. Can Random Forests, built on features extracted from traditional market data, predict the closing Sharpe ratio of short one-week delta-hedged straddles on EUR/USD, and do features derived from financial news articles improve the performance of the model?
2. Can Random Forest predictions be used to avoid bad trades and therefore improve the original systematic strategy?

4.1. Machine learning setup

This study aims to predict the closing Sharpe ratio of individual short one-week delta-hedged straddles. To this end, we constructed two different target variables. Given the sale of a straddle on trading day t and its closing date $t + 5$, target variable y_1 measures whether the position closed with a positive Sharpe ratio. Target variable y_2 measures whether the position closed with a weaker Sharpe ratio than minus one. Both are binary target variables and are respectively constructed as:

$$y_1(t) = \begin{cases} 1, & \text{if } \textit{Sharpe Ratio}(t+5) > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

$$y_2(t) = \begin{cases} 1, & \text{if } \textit{Sharpe Ratio}(t+5) < -1. \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

where *Sharpe Ratio* $(t + 5)$ denotes the closing Sharpe ratio of the short one-week delta-hedged straddle opened on trading day t and closed on trading day $t + 5$.

The target variables were predicted using Random Forest classifiers built with the Python package scikit-learn.[26] In total, we tried 120 different model configurations where each Random Forest was built with 1000 trees and a unique combination of hyperparameters that control maximum tree depth, the minimum number of samples required to be in a leaf node, and the maximum number of random features sampled for tree construction. Table 4 specifies the Random Forest configurations considered in this study.

Table 4
This table presents the different possible values considered for different hyperparameters available in the Random Forest implementation of scikit-learn. The default value is used for hyperparameters that are not listed.

Hyperparameter	Values
*n*_estimators	{1000}
max_depth	{None, 6, 8, 10, 12, 14}
min_samples_split	{2}
min_samples_leaf	{1, 10, 25, 50}
random_state	{42}
bootstrap	{True}
max_features	{1, 10, 25, 50}

Table 5
This table shows an example of walk-forward validation where t_i represents the feature vector of trading day i. Here, a training window of size four is taken (underlined), together with a test window of size one (boldfaced). The last element of the training set is consistently removed (slashed), leaving three feature vectors for training. This process is repeated j times where, after each iteration, the sliding window is shifted by one trading day.

Iteration	Variable roles
1	$\underline{t_1\ t_2\ t_3\ \cancel{t_4}}\ \mathbf{t_5}\ t_6\ \cdots\ t_n$
2	$t_1\ \underline{t_2\ t_3\ t_4\ \cancel{t_5}}\ \mathbf{t_6}\ \cdots\ t_n$
⋮	⋮
j	$t_1\ \cdots\ \underline{t_{n-4}\ t_{n-3}\ t_{n-2}\ \cancel{t_{n-1}}}\ \mathbf{t_n}$

The models were evaluated using walk-forward validation, which is a cross-validation technique designed specifically for temporally ordered data. It constructs train-test splits by partitioning the data chronologically such that the training data strictly precedes testing data. More concretely, for the first split, a model is built on a training window of m consecutive days where each day $t_{training} \in [t_0, t_{m-1}]$. After, its predictions are evaluated on a test window of n consecutive days where each day $t_{test} \in [t_m, t_{m+n-1}]$. Each following split shifts the training window forward by n days, after which the process is repeated. In our case, we chose a training window of $m = 5*252$ (five trading years) and a testing window of $n = 1*252$ (one trading year). This resulted in 16 out-of-sample trading years for evaluation. To avoid data leakage, we removed the last five trading days in each training window. Indeed, our target variable on day t uses information that is only available on day $t + 5$. Not removing this data would introduce a dependency between the train and test data, resulting in overoptimistic performance estimates. The evaluation method is illustrated in Table 5. We evaluated built models using accuracy and area under the receiver operating characteristic curve (ROC-AUC).

4.2. Ablation study

We performed an ablation study to investigate both the accuracy of the Random Forest predictions, and to what extent different feature sources contributed to the prediction performance. In total, five different feature combinations were considered, summarised in Table 6.

We applied our proposed machine learning setup to each feature matrix and averaged the out-of-sample performances of the different hyperparameter configurations. In addition to comparing the performances of the different feature combinations, the predictions were also compared to those of a stratified dummy classifier that randomly predicts based on the target variable distribution found in the training set. We performed this process for each target variable separately.

4.3. Strategy development

In contrast to the original baseline strategy S_0, which sells a new straddle each trading day, our strategies will only open a new position whenever the daily model prediction meets a certain criterion. In what follows, we outline how we

Table 6
This table lists the five different feature matrices considered in our ablation study, together with their dimensions.

Matrix	Features	Dimensions
X_1	market, time, and trading	$T \times 10$
X_2	X_1 and temporal features	$T \times 50$
$X_3^{(10)}$	X_2 and news features with $K = 10$	$T \times 235$
$X_3^{(20)}$	X_2 and news features with $K = 20$	$T \times 285$
$X_3^{(40)}$	X_2 and news features with $K = 40$	$T \times 385$

prepared and used the predictions from our Random Forests. This process breaks down into three parts: model selection, model probability calibration, and trade criterion selection.

First, we chose to use two different Random Forests in order to assess the added value of our news features for trading. One was built on feature matrix X_2, and one was built on $X_3^{(k)}$ and therefore uses news features. The optimal Random Forest and LDA hyperparameter configuration was selected based on the best ROC-AUC performance in the out-of-sample period spanning the years 2005 through 2009.

Second, we introduced probability calibration for our Random Forests. Although these classifiers output a real number between zero and one, it rarely represents a probability. For example, when a value of 0.8 is predicted, it does not correspond to an 80% chance of event occurrence. To obtain more accurate probability estimates, we trained model calibrators using in-sample Random Forest predictions and the CalibratedClassifier function from scikit-learn. The calibrators were trained using Platt's scaling,[27] and cross-validation with four traditional folds that were not shuffled to maintain temporal order. Out-of-sample predictions were then mapped to probabilities by applying the trained calibrator to the original Random Forest prediction.

Third and last, we developed two types of trading strategies S_1 and S_2 that respectively use models trained on target variable y_1 and y_2. Given trading day t, strategy S_1 opens a new position only if the model predicts a greater than 50% chance that the position will end with a positive Sharpe ratio. Strategy S_2 opens a new position only if the model predicts a smaller than 15% chance that the position will end up with a worse Sharpe ratio than minus one.

Both thresholds were chosen based on intuition and performance in the out-of-sample period spanning the years 2005 through 2009. We did not optimize the selected thresholds further. Each strategy comes in three variants: $S^{(market)}$ uses the predictions based on feature matrix X_2, $S^{(news)}$ is based on feature matrix $X_3^{(k)}$, and $S^{(both)}$ combines both. In the last case, a position will only be opened if the predictions of both models satisfy the strategy criterion. All this resulted in six new trading strategies which were evaluated in the out-of-sample period spanning the years 2010 through 2020. Table 7 summarizes the different trading strategies.

5. Results and discussion

This section shows and discusses the results obtained in this study. Section 5.1 first covers the ablation study, after which Section 5.2 reports on trading strategy development.

5.1. Ablation study results

The results of the ablation study for both target variables are showcased in Table 8. Each entry respectively denotes the mean and standard deviation of yearly performances during the out-of-sample period spanning the years 2005 through 2020. Note that one year corresponds to one test window (Section 4.1) and that the mean and standard deviation were computed on the average performance per trading year, which in turn was obtained by averaging the performances of the individual Random Forest configurations for that year.

First and foremost, we beat the dummy classifier for each feature setting and target variable. This implies that we indeed are able to predict the performance of our trades. Second, the generated temporal features used in feature setting X_2 seem to improve performance compared to feature setting X_1. Third, news features seem to increase performance even further. We note that different values for K seem to have little effect. For this reason, we expanded our ablation study to also include scenarios for $K = 5$ and $K = 15$. The conclusion remains largely the same and can be consulted in C. We only consider the scenario using $K = 20$ for the remainder of this section. The same patterns in improvement can be observed for both target variables. Note that absolute performance for y_2 is noticeably higher than for y_1, which is due to the more skewed class distribution evidenced by the dummy performance. Fourth and last, the standard deviations

Table 7
This table summarizes the three different trading strategies. Note that S_1 and S_2 have three variants based on the prediction model used.

Strategy	Description	Variants
S_0	Sells a new straddle every day	1
S_1	Sells a new straddle on days where predicted $\hat{y}_1 > 0.5$	3
S_2	Sells a new straddle on days where predicted $\hat{y}_2 < 0.15$	3

Table 8
This table shows the ablation study results (in pct) for both target variables, where each entry respectively denotes the mean and standard deviation of the yearly performances obtained during the out-of-sample period spanning the years 2005 through 2020.

		Dummy	X_1	X_2	$X_3^{(10)}$	$X_3^{(20)}$	$X_3^{(40)}$
Y_1	Acc	(60.0, 0.1)	(66.3, 5.1)	(69.2, 4.5)	(72.0, 4.1)	(72.2, 4.2)	(72.3, 4.5)
	AUC	(50.0, 0.0)	(61.1, 3.6)	(61.1, 4.2)	(62.7, 4.1)	(63.0, 4.4)	(62.9, 4.0)
Y_2	Acc	(69.0, 0.1)	(75.0, 4.6)	(77.0, 3.4)	(79.5, 3.2)	(79.6, 3.3)	(79.7, 3.3)
	AUC	(50.0, 0.0)	(66.2, 4.9)	(67.0, 5.0)	(68.0, 4.3)	(68.1, 4.5)	(67.4, 4.8)

seem to suggest that performance fluctuates across years. Fig. 1 shows the ROC-AUC for target variable y_1 and feature matrices X_2 and $X_3^{(20)}$ for all Random Forest configurations per out-of-sample period. The horizontal line in red indicates the stratified dummy classifier performance.

The performance improvement obtained by using news features seems to vary per year but is statistically significant according to the Wilcoxon rank-sum test applied to the difference of the medians (p-value $\ll 0.01$).[28] In total, we obtained a noticeable improvement for 12 out of 16 years. The improvement is most apparent in the last four years. In 2014, neither feature setting was able to beat the dummy classifier. More research is necessary to explain these phenomena. For y_2, news features yielded an improvement for 10 out of 16 years. In contrast to y_1, performance does not deteriorate in 2014. The ROC-AUC through time for y_2 is visually presented in D.

In an effort to understand the individual contribution of the counting, LDA, and lexicon-based news measures outlined in Table 2, we compared their individual performance to the combined approach ($X_3^{(20)}$). The results are displayed in Table 9. Note that each scenario also includes all features from X_2.

These results suggest that both counting ($X_{counting}$) and topic features ($X_{LDA}^{(20)}$) alone are sufficient to beat the classifier using only temporal market features (X_2). This does not seem to hold for the sentiment and subjectivity measures ($X_{lexicon}$). Several factors might cause sentiment features to under-perform in our application. First, we did not employ a lexicon catered towards financial language. Methods using more domain-specific lexicons might improve performance (e.g. [25,29]). Second, although our sentiment extractor exploits patterns such as negations, it still remains a very simple approach. More advanced techniques based on machine learning might yield better sentiment features (e.g.30). Moreover, our approach might measure sentiment inaccurately because it aggregates the sentiment of all confounding topics present in news articles. A fine-grained approach might therefore be more suitable (e.g. [31,32]). Third and last, defining sentiment for a currency pair is harder than for a stock. For example, 'strong dollar' is considered a positive sentence on EUR/USD by our extractor, whereas in reality it implies a negative outlook for its price rate. Ultimately, combining all alternative feature sources consistently yields the best performance.

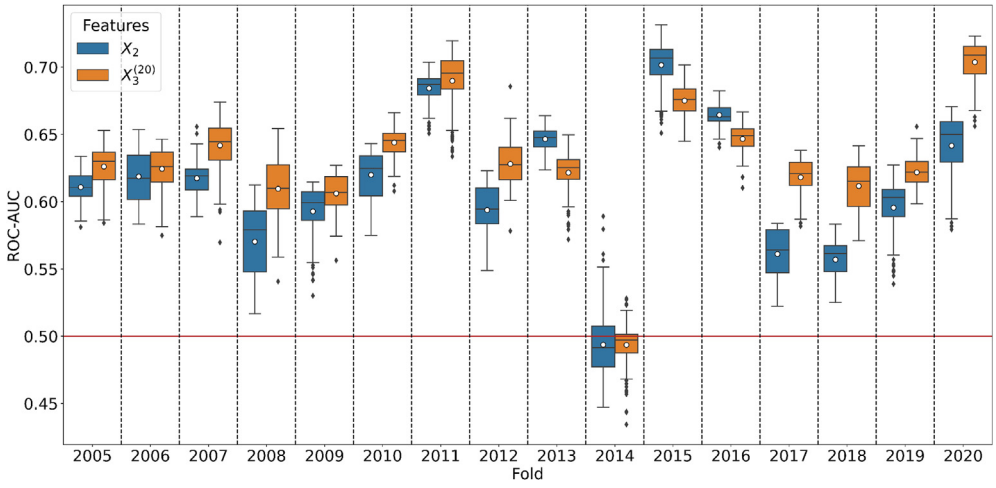

Fig. 1. This figure shows the ROC-AUC for target variable y_1 and feature matrices X_2 and $X_3^{(20)}$ for all Random Forest configurations per out-of-sample period.

Table 9
This table shows the results (in pct) of individual alternative feature sources and compares them to X_2 and $X_3^{(20)}$. Each entry respectively denotes the mean and standard deviation of the yearly performances obtained during the out-of-sample period spanning years 2005 through 2020.

		X_2	$X_{counting}$	$X_{LDA}^{(20)}$	$X_{lexicon}$	$X_3^{(20)}$
Y_1	Acc	(69.2, 4.5)	(71.8, 4.3)	(71.3, 4.2)	(69.4, 4.5)	(72.2, 4.2)
	AUC	(61.1, 4.2)	(62.2, 4.3)	(62.1, 4.4)	(61.1, 4.3)	(63.0, 4.4)
Y_2	Acc	(77.0, 3.4)	(79.0, 3.1)	(78.8, 3.2)	(77.3, 3.4)	(79.6, 3.3)
	AUC	(67.0, 5.0)	(67.7, 4.3)	(67.5, 4.3)	(67.0, 4.8)	(68.1, 4.5)

5.2. Trading strategy analysis

This section reports on the results obtained by applying our machine learning setup in a trading setting as outlined in Section 4.3. In what follows, we first address the results from our model selection and calibration procedure, after which we report on the results obtained by our proposed trading strategies.

5.2.1. Model selection and Calibration

We used four different Random Forest configurations for our trading strategies. They were selected based on the best average ROC-AUC performance during the out-of-sample period spanning the years 2005 through 2009. Table 10 shows the optimal hyperparameter configuration found for each target variable and feature setting. Remarkably, optimal hyperparameter configurations differ very little across target variables and feature settings.

Next, we evaluated the quality of the probability estimation using calibration curves.[33] Here, the predicted probability estimates are binned and then the fraction of true positives is computed per bin. Better calibrated estimates will be closer to the main diagonal in the plot. A model over (under) estimates probabilities if points fall below (above) the diagonal. Fig. 2 displays the calibration plot for the Random Forest using feature matrix $X_3^{(20)}$ for target variables y_1 and y_2. The results were obtained from out-of-sample predictions made in the years spanning 2005 through 2009.

Calibration is effective for both target variables. Compared to the original models, shown in red, the calibrated models, shown in blue, result in more accurate probability estimates. Note that the accuracy of probability estimates for y_2 is worse for prediction values that occur less often. This makes sense as in these cases the calibrator had less data to learn from. Results for the model using feature matrix X_2 are similar and are provided in E.

5.2.2. Trading strategies

We evaluated the proposed trading strategies on the out-of-sample period spanning years 2010 through 2020 and examined five different metrics: the number of days a new straddle was sold, the hit rate, and the mean, standard deviation, and fifth percentile of the closing Sharp ratios of sold straddles. The results are shown in Table 11. Note that we did not take transaction costs and slippage into account. Moreover, we assume that daily predictions were available at 5:00 pm ET, after which immediately a trading decision was made.

We notice four interesting outcomes. First, all proposed strategies and their variants do better than the baseline S_0 on all performance metrics. Trades have a higher expected Sharpe ratio, and the tail risk seems to be reduced. Second,

Table 10
This table shows the optimal Random Forest hyperparameter configuration for both target variables, based on obtained ROC-AUC during the out-of-sample period spanning the years 2005 through 2009.

Hyperparameter	Y_1		Y_2	
	X_2	$X_3^{(20)}$	X_2	$X_3^{(20)}$
n_estimators	1000	1000	1000	1000
max_depth	6	8	6	6
min_samples_split	2	2	2	2
min_samples_leaf	50	50	10	50
random_state	42	42	42	42
bootstrap	True	True	True	True
max_features	0.75	0.75	0.75	0.5

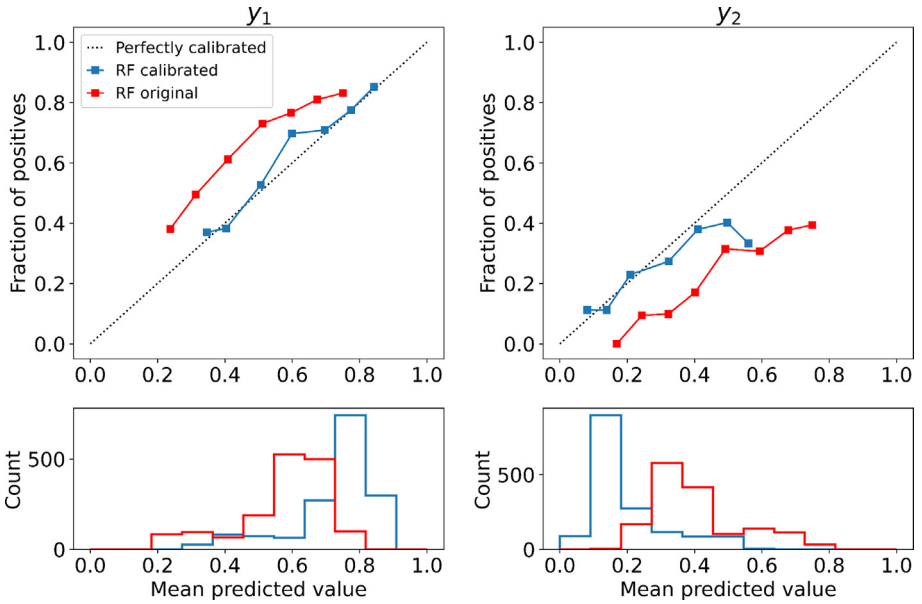

Fig. 2. This figure respectively shows the calibration plot for target variable y_1 and y_2 for the model using feature matrix $X_3^{(20)}$, together with a histogram of predicted values. The results were obtained from out-of-sample predictions made in the period spanning years 2005 through 2009.

Table 11
This table contains performance metrics for three different trading strategies: S_0, S_1, and S_2. The superscripts *market*, *news*, and *both* respectively denote a strategy that uses a model trained on market features (X_2), one that additionally uses news features ($X_3^{(20)}$), and one that combines both models. The results were obtained out-of-sample for the years spanning 2010 through 2020.

	Trades	Hit Rate (%)	Sharpe Ratio		
			μ	σ	$P_{5\%}$
S_0	2870	70.1	1.06	2.50	−3.08
$S_1^{(market)}$	2757	72.5	1.20	2.29	−2.68
$S_1^{(news)}$	2714	73.1	1.26	2.25	−2.48
$S_1^{(both)}$	2694	73.4	1.28	2.22	−2.48
$S_2^{(market)}$	1187	78.5	1.65	1.90	−1.88
$S_2^{(news)}$	1743	78.9	1.68	1.88	−1.83
$S_2^{(both)}$	1057	80.2	1.78	1.81	−1.65

strategy two seems much more conservative than the other approaches: it places fewer trades, but those placed perform better. Third, strategies based on news features seem to improve performance. Moreover, if the number of trades is considered, strategy $S_2^{(news)}$ performs notably better as it places 60% more trades of matching quality compared to its counterparts. Finally, combining the predictions from both models seems to improve performance even further. The two different Random Forests do not always agree, and combining their decision makes for a more conservative strategy that seems to also improve trade performance. Note that all derived strategies place fewer trades than the original one. This will also reduce the impact of slippage and transaction costs in practice.

6. Conclusion

Starting from a trading strategy that harvests the EUR/USD volatility risk premium by selling one-week straddles every weekday, we have successfully introduced an approach that can more skillfully determine when new trades should be placed to avoid unfavorable ones. More specifically, we draw three main conclusions from our results. First, the closing Sharpe ratio of short one-week delta-hedged straddles can be predicted to a certain extent using Random Forests built on traditional market features. Second, prediction accuracy can be further improved by quantifying

temporal information residing in time-series features, and by using news indicators, such as attention, entropy, and latent topics derived through different text processing techniques. Third, calibrating the probability estimates of the Random Forests enables exploiting their output to design intuitive new trading strategies. These novel strategies outperform the original one. Ultimately, our work demonstrates the feasibility of machine learning and alternative data for the improvement and development of trading strategies.

Declaration of competing interest

None.

Appendix A. Refinitv RCS Qcodes

Refinitiv adds various types of meta-information to each news item in the form of RCS Qcodes. In this study we used several of these codes to filter our initial corpus. Table A.1 displays the meaning of each code mentioned in Section 3.1.

Table A.1
This table presents the different Refinitv RCS Qcodes used in our study together with their explanation.

RCS Qcode	Explanation
A:9	Currencies and Foreign Exchange Markets
A:8	Money Markets
A:*n*	National Government Debt
A:2	Debt and Fixed Income Markets
E:A	Interest Rates and Policy
E:B	Monetary and Fiscal Policy, and Policy Makers
E:5	Economic News
E:C	Workforce
E:9	Economic Indicators
E:4 S	Currency Intervention
G:6 J	United States
G:B4	Euro Zone as a Whole
G:AL	Euro Zone
G:3	Western Europe
M:Y	US Federal Reserve
M:K	European Union
M:I	European Central Bank
M:E9	Government Finances

Appendix B. Topic modelling using Gensim

We used the Gensim implementation of Latent Dirichlet Allocation for our study. Table B.1 shows the employed hyperparameter configuration. Note that we filtered our dictionary by removing words that were present in more than 50% of the documents, or in less than 20.

Table B.1
This table presents the used hyperparameter configuration for the Gensim Latent Dirichlet Allocation implementation. The default value is used for hyperparameters that are not listed.

Hyperparameter	Values
num_topics	{10, 20, 40}
Chunksize	500
Passes	10
Iterations	50
random_state	42

For illustration purposes, Table B.2 shows the three most prominent topics obtained by an LDA model using $K = 20$ during four different periods (our study uses 16 periods in total). As the extracted latent topics are not directly human-interpretable, topics are represented by their five most important words accompanied by a rudimentary best-effort interpretation. Note how the prominent latent topics change through time, reflecting changes in news themes (e.g. Iraq war in 2003, the 2008 recession, the Greek debt crisis in 2010, and the China–United States trade war in 2018).

Table B.2
This table outlines the three most prominent latent topics found by the LDA methodology using $K = 20$ for four different periods spanning the years 2000 through 2020. Each row denotes a different latent topic represented by its five most important words, followed by a rudimentary interpretation based on said keywords.

Period	Keywords	Interpretation
2000	euro, ecb, european, bank, trichet	European Central Bank
|	fed, reserves, rate, repurchase, banking	Federal Reserve
2005	oil, prices, iraq, energy, war	Iraq War
2005	rate, lending, euros, liquidity, money	Central Banks
|	mortgage, house, treasury, mae, freddie	Subprime Mortgage Crisis
2010	inflation, recession, recovery, economy	Economic Climate
2010	euro, eu, greece, germany, bailout	Greek Debt Crisis
|	liquidity, rate, allotment, tender, operation	Central Banks
2015	economy, growth, crisis, risk, economic	Economic Climate
2015	fed, rates, interest, policy, inflation	Federal Reserve
|	european, government, euro, eu, debt	European Central Bank
2020	china, dollar, global, trade, currency	China-US Trade War

Appendix C. Extended ablation study

Following the results outlined in Section 5.1, we extended our study with two additional scenarios using $K = 5$ and $K = 15$. The results are outlined in Table C.1.

Table C.1
This table shows the ablation study results (in pct) for feature setting X_3 and for both target variables. Each entry respectively denotes the mean and standard deviation of the yearly performances obtained during the out-of-sample period spanning years 2005 through 2020.

		$X_3^{(5)}$	$X_3^{(10)}$	$X_3^{(15)}$	$X_3^{(20)}$	$X_3^{(40)}$
Y_1	Acc	(71.8, 4.3)	(72.0, 4.1)	(72.2, 4.2)	(72.2, 4.2)	(72.3, 4.5)
	AUC	(62.6, 4.2)	(62.7, 4.1)	(63.0, 4.2)	(63.0, 4.4)	(62.9, 4.0)
Y_2	Acc	(79.2, 3.3)	(79.5, 3.2)	(79.5, 3.1)	(79.6, 3.3)	(79.7, 3.3)
	AUC	(67.6, 4.5)	(68.0, 4.3)	(68.0, 4.3)	(68.1, 4.5)	(67.4, 4.8)

These results suggest that the optimal value for K lies somewhere between $K = 5$ and $K = 40$. We used $K = 20$ for our trading strategy implementation.

Appendix D. Yearly performance target variable y_2

Figure D.1 shows the ROC-AUC for target variable y_2 and feature matrices X_2 and $X_3^{(20)}$ for all Random Forest configurations per out-of-sample period. The horizontal line in red indicates the stratified dummy classifier performance. Note that the performance improvement obtained by using news features seems to vary per year but is statistically significant according to the Wilcoxon rank-sum test applied to the difference of the medians (p-value $\ll 0.01$).

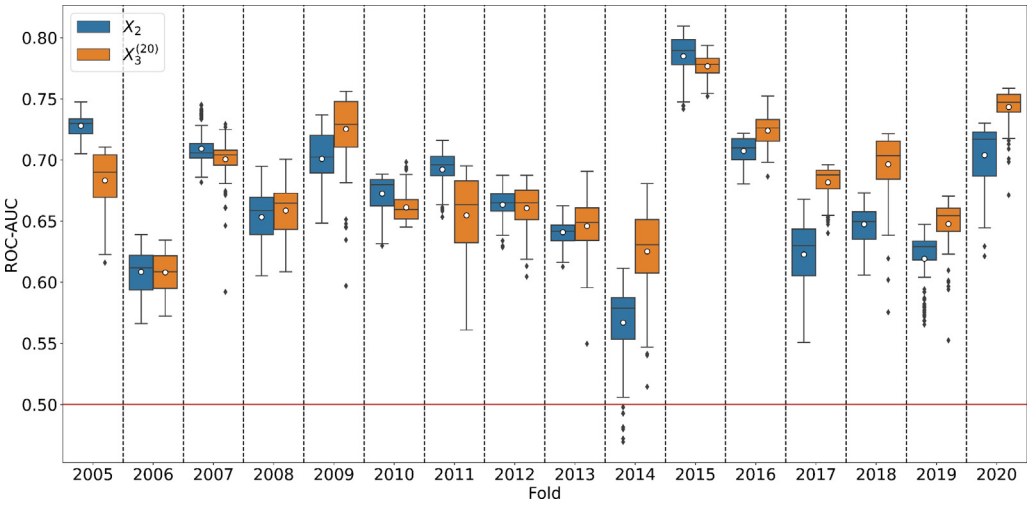

Fig. D.1. This figure shows the ROC-AUC for target variable y_2 and feature matrices X_2 and $X_3^{(20)}$ for all Random Forest configurations per out-of-sample period.

Appendix E. Calibration plot feature matrix X_2

Figure E.1 displays the calibration plot for the Random Forest using feature matrix X_2 for respectively target variable y_1 and y_2. The results were obtained from out-of-sample predictions made in the period spanning years the 2005 through 2009.

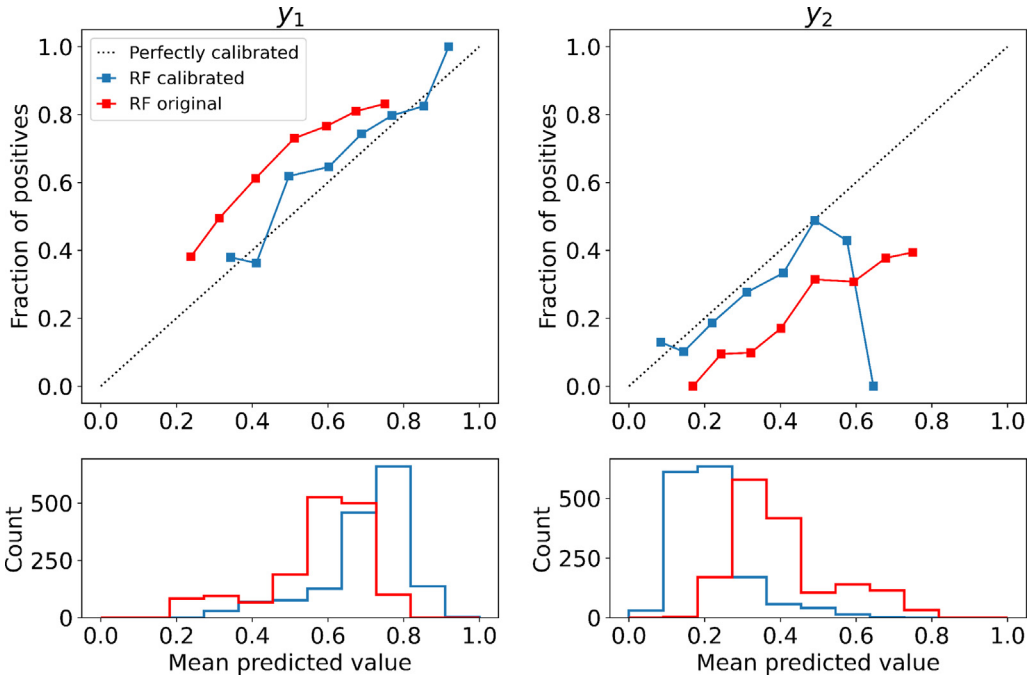

Fig. E.1. This figure respectively shows the calibration plot on target variable y_1 and y_2 for the model using feature matrix X_2, together with a histogram of predicted values.

References

1. Carr P, Wu L. Variance risk premiums. *Review of Financial Studies*. 2009;22:1311–1341. https://doi.org/10.1093/rfs/hhn038.
2. Dapena JP, Siri JR. *Index Options Realized Returns Distributions from Passive Investment Strategies*. ERN: Asset Pricing Models (Topic); 2015. https://doi.org/10.2139/ssrn.2733774.
3. Bondarenko O. An analysis of index option writing with monthly and weekly rollover. *SSRN Electronic Journal*. 2016. https://doi.org/10.2139/ssrn.2750188.
4. Schulte D, Stamos M. The performance of equity index option strategies during the financial crisis* the performance of equity index option strategies during the financial crisis. *SSRN Electronic Journal*. 2015. https://doi.org/10.2139/ssrn.2669999.
5. Daviaud O, Korber O, Mukhopadhyay A, Ungari S. *Systematic Trading in Options*. Société Générale - Cross Asset Research; 2020.
6. Checkley M, Higón DA, Alles H. The hasty wisdom of the mob: how market sentiment predicts stock market behavior. *Expert Systems with Applications*. 2017;77:256–263. https://doi.org/10.1016/j.eswa.2017.01.029.
7. Oliveira N, Cortez P, Areal N. The impact of microblogging data for stock market prediction: using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*. 2017;73:125–144. https://doi.org/10.1016/j.eswa.2016.12.036.
8. Curme C, Zhuo YD, Moat HS, Preis T. Quantifying the diversity of news around stock market moves. *Journal of Network Theory in Finance*. 2015;3:1–20. https://doi.org/10.21314/JNTF.2017.027.
9. Feuerriegel S, Ratku A, Neumann D. Analysis of how underlying topics in financial news affect stock prices using latent dirichlet allocation. In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. 2016:1072–1081. https://doi.org/10.1109/HICSS.2016.137.
10. Feuerriegel S, Pröllochs N. Investor reaction to financial disclosures across topics: an application of latent dirichlet allocation. *Decision Sciences*. 2018;52. https://doi.org/10.1111/deci.12346.
11. Theil K, Stajner S, Stuckenschmidt H. Word embeddings-based uncertainty detection in financial disclosures. In: *ECONLP@ACL*. 2018. https://doi.org/10.18653/v1/W18-3104.
12. Garman MB, Kohlhagen SW. Foreign currency option values. *Journal of International Money and Finance*. 1983;2(3):231–237. https://doi.org/10.1016/S0261-5606(83)80001-1.
13. Black F, Scholes M. The pricing of options and corporate liabilities. *Journal of Political Economy*. 1973;81(3):637–654. https://doi.org/10.1086/260062.
14. Sepp A. When you hedge discretely: optimization of sharpe ratio for delta-hedging strategy under discrete hedging and transaction costs. *SSRN Electronic Journal*. 2012. https://doi.org/10.2139/ssrn.1865998.
15. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. *Journal of Machine Learning Research*. 2003;3:993–1022. https://doi.org/10.5555/944919.944937.
16. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32. https://doi.org/10.1023/A:1010933404324.
17. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks; 1984. https://doi.org/10.1201/9781315139470.
18. Khoshgoftaar TM, Van Hulse J, Napolitano A. Comparing boosting and bagging techniques with noisy and imbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*. 2011;41(3):552–568. https://doi.org/10.1109/TSMCA.2010.2084081.
19. Sharpe WF. The sharpe ratio. *The Journal of Portfolio Management*. 1994;21(1):49–58. https://doi.org/10.3905/jpm.1994.409501. arXiv:https://jpm.pm-research.com/content/21/1/49.full.pdf.
20. Honnibal M, Montani I. *spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing, to appea*. 2017. r.
21. Rehurek R, Sojka P. *Gensim–python Framework for Vector Space Modelling*vol. 3. Brno, Czech Republic: NLP Centre, Faculty of Informatics, Masaryk University; 2011.
22. Shannon CE. A mathematical theory of communication. *The Bell System Technical Journal*. 1948;27(3):379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.
23. Loria S. Textblob documentation. *Release 0*. 2018;15 2.
24. Esuli A, Sebastiani F. Sentiwordnet: a publicly available lexical resource for opinion mining. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. European Language Resources Association (ELRA); 2006.
25. Loughran T, McDonald B. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*. 2011;66(1):35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x.
26. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*. 2011;12:2825–2830.
27. Platt J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.* 2000;10.
28. Demšar J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*. 2006;7:1–30.
29. Barbaglia L, Consoli S, Manzan S, Tiozzo Pezzoli L, Tosetti E. *Sentiment Analysis of Economic Text: A Lexicon-Based Approach, working paper available at: SSRN 4106936*. 2022.
30. Akhtar MS, Ekbal A, Cambria E. How intense are you? predicting intensities of emotions and sentiments using stacked ensemble [application notes]. *IEEE Computational Intelligence Magazine*. 2020;15(1):64–75. https://doi.org/10.1109/MCI.2019.2954667.
31. Dridi A, Atzeni M, Reforgiato Recupero D. Finenews: fine-grained semantic sentiment analysis on financial microblogs and news. *International Journal of Machine Learning and Cybernetics*. 2019;10. https://doi.org/10.1007/s13042-018-0805-x, 08.

32. Consoli S, Barbaglia L, Manzan S. Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowledge-Based Systems*. 2022;247, 108781. https://doi.org/10.1016/j.knosys.2022.108781.
33. Niculescu-Mizil A, Caruana R. Predicting good probabilities with supervised learning. In: *Proceedings of the 22nd International Conference on Machine Learning*. 2005:625–632. https://doi.org/10.1145/1102351.1102430.