UNIVERSITY OF WARSAW | FACULTY OF ECONOMIC SCIENCES

WORKING PAPERS
No. 13/2023 (420)

FROM ALCHEMY TO ANALYTICS: UNLEASHING THE POTENTIAL OF TECHNICAL ANALYSIS IN PREDICTING NOBLE METAL PRICE MOVEMENT

MARCIN CHLEBUS
ARTUR NOWAK

WARSAW 2023

WORKING PAPERS 13/2023 (420)

From Alchemy to Analytics: Unleashing the Potential of Technical Analysis in Predicting Noble Metal Price Movement

Marcin Chlebus, *Artur Nowak*

University of Warsaw, Faculty of Economic Sciences
Corresponding authors: mchlebus@wne.uw.edu.pl, artur.c.nowak@gmail.com

Abstract: Algorithmic trading has been a central theme in numerous research papers, combining knowledge from the fields of Finance and Mathematics. This thesis aimed to apply basic Technical Analysis indicators for predicting price movement of three noble metals: Gold, Silver, and Platinum in a form of multi-class classification. That task was performed using four algorithms: Logistic Regression, k-Nearest Neighbors, Random Forest and XGBoost. The study incorporated feature filtering methods such as Kendall-tau filtering and PCA, as well as five different data frequencies: 1, 5, 10, 15 and 20 trading days. From a total of 40 potential models for each metal, the best one was selected and evaluated using data from period 2018-2022. The result revealed that models utilizing only Technical Analysis indicators were able to predict price movements to a significant extent, leading to investment strategies that outperformed the market in two out of three cases.

Keywords: precious metals, algotrading, machine learning, multiclass classification, logistic regression, nearest neighbors, random forest, xgboost

JEL codes: C38, C51, C52, C58, G17

1 Introduction

Flight-to-quality is a process of rebalancing investors' portfolios to less risky assets in times of economic uncertainty, noticed first by Bernanke et al (1994). One type of those safer assets are precious metals. They have been appreciated for centuries for their scarcity, beauty and practical applications, e.g., as a store of value, due to their limited supply. Noble metals, especially Gold and Silver are widely considered as safe-haven assets during times of uncertainty and inflation hedge because they are thought to be more stable and less susceptible to economic fluctuations experienced by more risky assets, such as corporate bonds or equities. (Baur and Lucey, 2010; Bampinas and Panagiotidis, 2015). Platinum is not a popular commodity among both investors and the scientific community. Few studies on Platinum metals show contradictory results, as Hood and Malik (2013) do not find Platinum as a safe haven, while a few years later McCown and Shaw (2017) showed otherwise.

There are several factors that affect the price of noble metals. Radetzki and Wårell (2020) point out that the global demand for them is driven by various factors. The demand for Gold is driven mostly by investment demand, jewelry production and central bank purchases, for Silver - by industrial application and as a role as a monetary metal and for Platinum - solely by its industrial applications. Radetzki and Wårell (2020) indicate Platinum, along with Chrome and rare-earth elements as prime examples of indispensable materials with few substitutes and thus very low-price elasticity of demand. Since those commodities are treated as an inflation hedge, their price is also influenced by the monetary policy of central banks. Changes in interest rates, currency weakening can cause a flight-to-quality effect.

On the other hand, the Efficient Market Hypothesis (Fama, 1970) suggests that all available information is already incorporated in the asset prices, making it nearly impossible to outperform the market consistently. More recent studies (Piasecki and Stasiak, 2020) however shows that financial markets are not always perfectly efficient and there exists an opportunity for investors to reach abnormal returns. Behavioral finance also challenges many claims of EMF and points at Technical Analysis as a tool to outperform the market (Dehnad, 2011).

Machine learning algorithms have transformed the Finance world by providing new tools for analyzing complex financial datasets and supporting business decision making. They can be used not only to forecast the stock/foreign exchange (Huang et al., 2005), but also for detecting financial fraud (Perols, 2011) or optimizing portfolio (Ban et al., 2018).

Contrary to simple strategies such as Buy & Hold or trend-following, Technical Analysis in combination with Machine Learning allows for usage of more complex trading systems that can leverage complex relationships between various market indicators (Teixeira and De Oliveira, 2010; Aguirre et al., 2020). Models can analyze large amounts of data and continuously find patterns and relationships that are not immediately apparent to the human eye.

The primary objective of this paper is to develop a ML-based investment strategy for trading gold-, silver- and platinum-dollar pairs, respectively: XAU/USD, XAG/USD, XPT/USDT on a daily interval. It is translated into a multi-class classification problem, with three classes BUY, SELL and DO-NOTHING. Classes are determined by the percentage change in the closing price of the subsequent 1, 5, 10, 15 or 20 trading days. The exact threshold varies depending on the specific pair and frequency. However, it is chosen to ensure as equal class distribution as possible. Suppose some threshold $t: t > 0$, then observations classified as class SELL have values $(-\infty, -t)$, as class DO-NOTHING: $[-t, t]$ and as class BUY: (t, ∞), while each of three classes contains approx. 33% of total observations.

The scope of this paper is limited to analyzing daily OLHC data and technical analysis (TA) indicators for the three pairs. The study will focus on the performance of various ML models: Logistic Regression, k-Nearest Neighbors, Random Forest and XGBoost through two aspects. The first factor is evaluation through traditional metrics such as Balanced Accuracy and F1-score. The second factor is profitability of the investment strategies resulting from generated signals in comparison to traditional investment method – Buy & Hold. Additionally, to build possibly the least complex model, two feature selection methods are applied: Kendall rank correlation coefficient and Principal Component Analysis. Such choice is justified by the fact that some TA indicators are highly correlated with each other. By eliminating redundant features that provide similar information about the data or combining them into Principal Components, the cardinality of the data is significantly reduced.

This paper aims to answer following research questions:

1. How do different ML algorithms perform in predicting the price movement of pairs XAU/USD, XAG/USD and XPT/USD in different time horizons: from 1 day to 1 month (20 trading days)?
2. Can Technical Analysis indicators alone, without creating mathematical rules, predict market movements in different time horizons?

3. Which model performs best in terms of classification evaluation metrics: Balanced Accuracy and weighted F1- score?
4. Is the developed investment strategy able to outperform a simple Buy and Hold strategy?

The remainder of the paper is organized as follows. The second part presents a review of the literature regarding ML applications in Finance, Technical Analysis and previous research on ML for investment strategies. The third part is devoted to data and methodology used in research. The fourth part describes results of the empirical research and answers research questions. The last part summarizes the findings, contributions and suggests future research directions.

2 Literature review

2.1 Traditional approaches to price prediction

2.1.1 Technical Analysis

Technical Analysis (TA) is a popular approach in forecasting asset prices by scrutinizing historical price and volume data to identify patterns and trends (Murphy, 1999). Although TA has demonstrated success in some instances, it is based on subjective interpretations and may not be consistently effective across different market conditions (Park and Irwin, 2007).

TA indicators can be used as explanatory variables in the ML model (Patel et al., 2015). Technical Analysis can be classified into two categories: charting and mechanical methods (Zarrabi et al., 2017). Charting is a classical method and uses historical price patterns to predict future movements. This approach is highly subjective and reliant on the analyst's interpretation, making it hard to implement in ML models, contrary to mechanical methods based solely on mathematical rules. The number of such rules is a matter of discussion in the scientific community. According to Qi and Wu (2006), selecting an insufficient number of rules can lead to bias in statistical inference due to data mining. Too many rules will greatly increase the cardinality of the data.

The foundation of TA lies in the belief that historical price movements are repetitive and by examining past trends future price fluctuations can be predicted (Pring, 2002). Despite its popularity, the effectiveness of TA is disputed by academics and practitioners. Some studies show that under certain market conditions, strategies relying on TA indicators can outperform the market and produce excess returns (Brok et al., 1992; Dehnad, 2021), while other studies have questioned the effectiveness of TA, portraying its success because of Data-snooping (Sullivan et al., 1999). Bajgrowicz and Scaillet (2012) argues that performance of Technical Trading, although outperforms the market, is completely offset by the introduction of even low transaction costs.

2.1.2 Fundamental Analysis

First assumptions of Fundamental Analysis have been presented by Graham and Dodd (1934). Contrary to TA, Fundamental Analysis does not rely on past price movements but rather examines various underlying factors, economic indicators, or industry trends. Studies regarding

FA in the context of noble metals focus mainly on Gold and highlight numerous factors that can be analyzed, i.e. interest rates, inflation or currency movements (Tulley and Lucey, 2007).

Interest rates play a significant role in the valuation of Gold. As interest rates rise, during times of economic shocks, the opportunity cost of holding non-interest assets, i.e. noble metals, increases leading to lower demand and potentially lowering prices. Baur and McDermott (2010) confirmed this in their study but also indicated that this relationship is regional, as in Australia, Canada, Japan and BRIC countries (large emerging markets), Gold is not a safe haven.

Historically, noble metals have been considered hedges against inflation due to their limited supply and intrinsic value (Feldstein, 1980). Inflation erodes the purchasing power of fiat currencies and as a result, investors turn to i.e. precious metals as a means of preserving wealth, driving up demand and prices (Baur and McDermott, 2010; Baur and Lucey, 2010; Coudert and Raymond, 2011).

Currency movements, especially the US dollar, can also influence prices of noble metals. Most of them, including the aforementioned: Gold, Silver and Platinum are priced in US dollars, thus depending on its movements. (Capie et al., 2005; Worthington and Pahlavani, 2007). Weaker dollar makes these assets more affordable for investors holding other currencies and stronger dollar - otherwise.

Radetzki (1989) points out that Silver and Platinum are different from Gold and different fundamental determinants drive their prices. It is mainly the influence of the industry due to usage of those two metals in various industry branches.

Effectiveness of this approach is limited by the availability and quality of relevant data. Inflation rate in the US is made publicly available once per month, while interest rates change at most once per month without any regularity. Such irregular data makes it difficult to build trading systems and according to Menkhoff's (2010) research, Technical Analysis continues to be widely used among practitioners, with many fund managers preferring it over Fundamental Analysis as a tool for market decision making.

2.2 Machine Learning in Finance

Credit scoring is a vital aspect of the financial industry and various ML models have been implemented for this purpose. Logistic Regression is a commonly used model due to its simplicity and transparency of predictions. More sophisticated algorithms can outperform simple LR, but they have their own flaw: incapability to explain predictions. There is no

agreement on the best-performing model. Dastile et al. (2020) conducted a survey analyzing commonly used models in credit scoring, comparing their performance on German and Australian credit data. The results showed that ensemble models generally outperform single classifiers and Neural Networks outperform standard statistical models.

Financial fraud is a crucial issue in corporate and finance business. One of the most common and potentially dangerous types of financial frauds is credit card fraud. It is also the type with the most literature on prevention and detection on it. Here ML-based systems outperform traditional econometric models as well (Popat and Chaundhary, 2018) but the usage of Black-Box models is limited in some parts of the world, including European Union, due to regional legislation on Explainable AI (MiFID II).

Portfolio optimization is a problem containing two stages of decision-making: selection of stocks/commodities and their distribution in the portfolio. Optimization models should use historical data to select stocks and assign portfolio proportions to them. Pareek and Thakkar (2015) point out that ML techniques are widely used by the researchers and accepted in the scientific community for analyzing stock market behavior and optimizing portfolio.

Analyzing financial data in general is a challenging task that demands the development of innovative and complex models. Traditional approaches such as ARIMA have been reported to represent data accurately. On the other hand, traditional models have been proven to be unsuitable for handling sparse datasets and identifying underlying relationships between variables (Rundo et al., 2019), contrary to ML algorithms.

2.3 Machine learning based Investment strategies

The task of predicting stock market behavior can be classified based on the type of output to be estimated, namely, classification and regression. The former translates the problem to return categories describing the future behavior, usually defined as UP or DOWN, while the latter involves numerical predictions of the extent to which a stock may increase or decrease. A survey conducted by Kumbure et al. (2022) on 138 articles published between 2000 and 2019 shows the majority (55%) of these articles use regression models to forecast financial markets, while classification problems make up only 44% and the remaining 1% are clustering models (Kumbure et al., 2022).

Authors show that the most popular technique across the 138 articles is simple Neural Network, especially in early studies (Kumbure et al., 2022). Several authors have highlighted

difficulties with Neural Networks, i.e., time complexity (Das and Padhy, 2018) or overfitting and suggest using Support Vector Machine for both classification and regression problems (Yeh et al., 2011; Li et al., 2016). Kumbure et al. (2022) also point out the existence of Ensemble models in this field - accounting for 5 articles.

Bustos and Pomares-Quimbaya (2019) conducted a similar survey among 53 articles from 2014 to 2018 focusing on stock market movement prediction, namely a classification problem. The most popular technique presented in 17 articles is Support Vector Machine, followed by Ensemble Classifiers (12 articles). Authors point out the increasing popularity of Neural Networks (10 articles) and Deep Learning (9 articles). Traditional methods such as Logistic Regression now serve as a benchmark for more sophisticated ML algorithms rather than a method itself (Huang and Li, 2017).

Feature selection is a method to reduce computational complexity of training the model and decrease the risk of over-fitting by removing variables carrying little to no predictive power. Researchers use one of three types of feature selection methods. The first are filter methods, involving selecting relevant features before model training, based on some pre-defined criteria (Barak and Modarres, 2015). The second are wrapper methods that iteratively select a subset of features, train the model, and then evaluate it based on a pre-defined metric (Zhang et al., 2014). The third one is a hybrid of filter and wrapper methods. Lee (2009) first used a filter-based method to filter the least relevant features and reduce the computational complexity of model training and then applied a wrapper-based method to select the optimal feature subset.

2.4 Research Gap and Contribution

Existing research has demonstrated the potential of Machine Learning algorithms in investment strategy development. However, there remains a gap, as most of the studies revolve around the stock market and limited studies focus on noble metals, even less scientific articles examine multi-class classification in this context.

The paper aims to fill this gap by developing and evaluating a ML-based investment strategy for XAU/USD, XAG/USD and XPT/USD pairs using Logistic Regression, k-Nearest Neighbors, Random Forest and XGBoost algorithms. Technical Analysis indicators will be utilized as solely input features for the models. Furthermore, this research will compare the performance of the developed models and the investment strategies built based on the best performing model. Research will also investigate the time horizon of the target variable:

whether to predict the price movement in the following day, 5 days (1 trading week), 10 days (2 trading weeks), 15 days (3 trading weeks) and 20 days (1 trading month).

The proposed approach also addresses the problem of an adequate feature selection algorithm. Some Technical Analysis indicators are correlated with each other, so two methods of handling multicollinearity in the dataset are introduced: Principal Component Analysis (Zhong and Enke, 2017; Singh and Srivastava, 2017) and removing highly correlated features.

3 Data and Methodology

3.1 Dataset

In this study, historical daily Open, High, Low and Close (OHLC) price data was acquired for three pairs correlated with the United States Dollar (USD) - Gold (XAU), Silver (XAG) and Platinum (XPT) - spanning from January 1, 2000, to December 31, 2022, as seen on Figure 1**.** Data was obtained from Stooq.pl database. Notably, contrary to the Stock Market, variable accounting for Volume is missing. It is attributed to the decentralized nature of the Foreign Exchange Market, where trading volumes are not consistently reported across various trading venues (Flood, 1994).

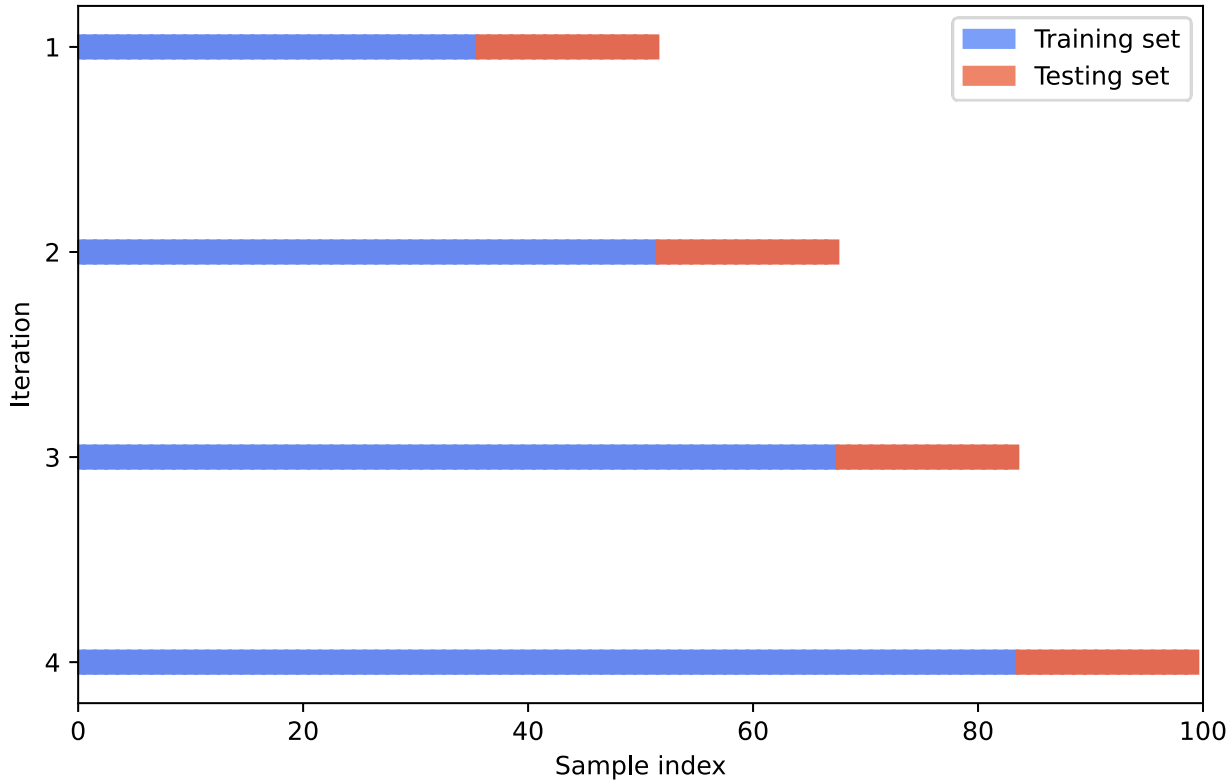
Figure 1: Price of XAU/USD (Gold), XAG/USD (Silver) and XPT/USD (Platinum) between 2000 and 2022

Source: Own calculations

The dataset used in this study is divided into two parts, namely a training set and a testing set. The training set spans from January 1, 2000, to December 31, 2017, while the testing set covers the period from January 1, 2018, to December 31, 2022. The division was applied not only to maintain an approximate 80:20 ratio between the training and the testing set but also to include pre- and post-pandemic periods. To evaluate and choose the best model for each noble metal,

a 4-fold Time Series cross-validation is employed on the training data, while preserving its time-dependency. Each fold, the model is trained on the past data and evaluated on the future data. Successive training sets are supersets of the preceding ones (Figure 2).

Figure 2 An example of 4-fold Time Series Cross Validation



Source: Own calculations

To obtain cross-validated scores, the metrics are calculated on the CV-testing sets for each subset and then averaged. This approach ensures that the models are trained on a diverse range of data and can generalize well to new, unseen data. The use of cross-validation also helps to prevent overfitting, where the models perform well on the training data but poorly on new, unseen data.

3.1.1 Feature Engineering

3.1.1.1 Independent variables

Using the *Technical Analysis Library* (Technical Analysis Library, 2018), a Python-based tool developed for financial market analysis, 73 technical indicators can be calculated from Open-High-Low-Close (OHLC) data. This library is built on top of efficient numerical computing libraries, like Pandas and NumPy, and provides a range of analytical tools that help in creating

and implementing various trading strategies. These tools allow market participants to make informed decisions based on historical price and volume data.

TAL's technical indicators are divided into four main groups: Momentum, Volume, Volatility, and Trend indicators. However, it's important to note that Volume indicators are not suitable for forex market data because of the market's decentralized nature. This decentralization leads to a lack of accurate volume information, making the use of 10 Volume indicators offered by *TAL* ineffective in this context. As a result, the analysis of Forex data mainly focuses on the other three categories of technical indicators - Momentum, Volatility, and Trend indicators (73 in total) - to develop effective trading strategies and investment models.

TAL offers indicators as follows:

- Momentum indicators: Awesome Oscillator, Kaufman's Adaptive Moving Average (KAMA), Percentage Price Oscillator (PPO), Percentage Volume Oscillator (PVO), Rate of Change (ROC), Relative Strength Index (RSI), Stochastic RSI, Stochastic Oscillator, True Strength Index (TSI), Ultimate Oscillator, Williams %R.
- Volatility indicators: Average True Range (ATR), Bollinger Bands, Donchian Channel, Keltner Channel, Ulcer Index.
- Trend indicators: Average Directional Movement Index (ADX), Aroon Indicator, Commodity Channel Index (CCI), Detrended Price Oscillator (DPO), Exponential Moving Average (EMA), Ichimoku Indicator, KST Oscillator, Moving Average Convergence Divergence (MACD), Mass Index (MI), Parabolic Stop and Reverse (Parabolic SAR), Schaff Trend Cycle (STC), Trix (TRIX), Vortex Indicator (VI), Weighted Moving Average (WMA), Average Directional Movement Index (ADX).

In contrast to the approaches found in the literature, such as those presented by Zarrabi et al. (2017) and Qi and Wu (2006), this study will not involve the development of specific mathematical rules derived from the technical indicators. The absence of such rules represents a deviation from the traditional methodology, which often attempts to create customized frameworks for market trend predictions based on the relationships between these indicators.

Instead, the focus of this research will be on utilizing the technical indicators themselves, without any additional transformations or derived mathematical constructs. The hope is placed in black-box models, such as Random Forest or XGBoost. This approach will emphasize the individual capabilities of the indicators and their potential in predicting market

trends. By relying solely on the indicators, the study aims to determine their inherent predictive power and evaluate their effectiveness in the context of market trend analysis.

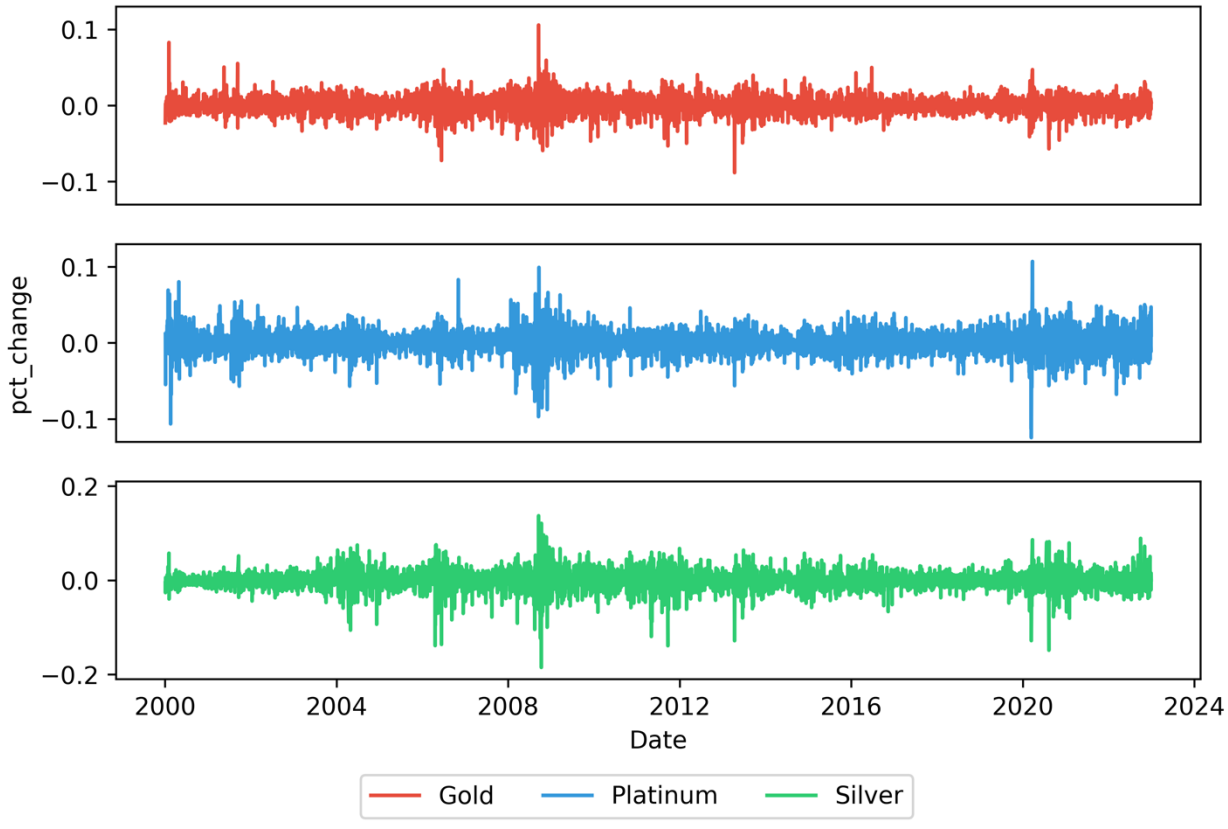
3.1.1.2 Dependent variable

In this context, the dependent variable is a three-level categorical variable that is defined based on the percentage difference between the closing prices of three metals - gold, silver, or platinum - over a period of n consecutive days:

$$y_{\text{metal}} = \begin{cases} -1 \iff \dfrac{Close_{t+1}}{Close_t} < -thresh_{metal:frequency} \\ 0 \iff -thresh_{metal} \leq \dfrac{Close_{t+1}}{Close_t} \leq thresh_{metal:frequency} \\ 1 \iff \dfrac{Close_{t+1}}{Close_t} > thresh_{metal:frequency} \end{cases} \quad (1)$$

The choice of using returns was based on their stationarity property, which is not present in prices. Returns also allow for a separation of the forecast from the current price and the analysis can focus on the direction and magnitude of price changes, rather than the absolute price levels.

Figure 3 Daily returns of XAU/USD (Gold), XAG/USD (Silver) and XPT/USD (Platinum) between 2000 and 2022.



Source: Own calculations

The variable is defined based on separate thresholds for each metal, which were selected to achieve a similar, approximately equal class distribution across the three metals for the BUY:DO-NOTHING:SELL classes. Calculated thresholds are presented on Figure 4.

Figure 4. Independent variable class distribution across XAU/USD (Gold), XAG/USD (Silver) and XPT/USD (Platinum)

Metal	Frequency	Thresh	BUY	DO-NOTHING	SELL
Gold	1D	0.35%	1947 (34.2%)	2015 (35.4%)	1728 (30.4%)
Gold	5D	0.91%	425 (36.9%)	398 (34.5%)	330 (28.6%)
Gold	10D	1.46%	202 (35.1%)	228 (39.6%)	146 (25.3%)
Gold	15D	1.57%	155 (40.4%)	118 (30.7%)	111 (28.9%)
Gold	20D	1.84%	101 (35.1%)	108 (37.5%)	79 (27.4%)
Silver	1D	0.57%	1921 (34.4%)	1960 (35.1%)	1705 (30.5%)
Silver	5D	1.37%	403 (35.0%)	400 (34.7%)	350 (30.4%)
Silver	10D	2.06%	200 (34.7%)	202 (35.1%)	174 (30.2%)
Silver	15D	2.89%	137 (35.7%)	147 (38.3%)	100 (26.0%)
Silver	20D	3.51%	95 (33.0%)	108 (37.5%)	85 (29.5%)
Platinum	1D	0.49%	1951 (34.5%)	1903 (33.7%)	1801 (31.8%)
Platinum	5D	1.12%	426 (36.9%)	378 (32.8%)	349 (30.3%)
Platinum	10D	1.59%	227 (39.4%)	171 (29.7%)	178 (30.9%)
Platinum	15D	2.08%	149 (38.8%)	119 (31.0%)	116 (30.2%)
Platinum	20D	2.7%	105 (36.5%)	96 (33.3%)	87 (30.2%)

Note: Metal – one of three metals, Frequency – time frequency of the data in trading days (D – trading day), Thresh – the absolute value of calculated thresholds that divides variable into one of three classes: SELL: $[-\infty; -thresh)$; DO-NOTHING: $[-thresh;\ thresh)$; BUY: $[thresh, \infty)$, BUY – number of observations of the BUY class (percentage share), DO-NOTHING – number of observations of the DO-NOTHING class (percentage share), SELL – number of observations of the SELL class (percentage share).

Source: Own calculations

3.1.2 Feature Selection

In the study, 73 technical indicators were computed and integrated into the dataset. These variables might exhibit high correlation as they are derived from the same underlying price data. To mitigate the effects of redundant features, two approaches were introduced:

- Implementing *Principal Component Analysis* (PCA), a dimensionality reduction technique that transforms the set of variables into a new set of uncorrelated linear combinations (Jolliffe and Cadima, 2016). Before applying PCA, data is scaled to the range between 0 and 1, because variables have different ranges and thus would distort the algorithm.
- Identifying and removing highly correlated features with Kendall's rank correlation coefficient, ensuring that the remaining variables provide information without

redundancy. The most common method across different fields of science is to select variables correlated $|r| < 0.7$ (Dormann et al. ,2013). Although in this study, Pearson's correlation coefficient is not used, that threshold will be used for filtering features, namely variables exhibiting $|\tau| < 0.7$ with other dependent variables.

3.1.2.1 Principal Component Analysis

PCA is an orthogonal linear transformation widely used as a dimensionality reduction technique that projects the original dataset onto a lower-dimensional space while retaining most of the data's variability (Jolliffe and Cadima, 2016). PCA transforms the original set of variables into a new set of uncorrelated linear combinations, capturing the maximum variance in the data. The first component is calculated to maximize the variance explained in the dataset and the subsequent components explain the maximum of the remaining variance. The transformed dataset usually consists of fewer dimensions than the original one, while retaining the essential information to a given extent.

Principal Component Analysis has numerous advantages, addressing various challenges associated with high cardinality of the data. One of the primary benefits is its ability to mitigate the impact of multicollinearity on model performance. Multicollinearity is understood as the presence of high correlation between predictor variables, which can lead to violation of statistical assumptions or model instability or inflation of standard errors and additionally it increases the difficulty of interpreting the importance of individual variables (Dormann et al., 2013). PCA effectively addresses that issue by transforming the original correlated variables into a new set of uncorrelated linear combinations, making the model more stable and reliable.

Another advantage is its ability to improve the model performance by reducing the computational complexity associated with high-dimensional data. In many algorithms, usage of sparse datasets containing many variables lead to increased computational costs and longer training time. By reducing the cardinality of data, PCA allows for faster model training and improved performance, enabling a more efficient decision-making process.

An additional advantage of PCA is the ability to enhance noise reduction in high-dimensional datasets. Many datasets, especially financial data, contain a noise which makes the identification of underlying patterns and trends harder thus impacting negatively on the performance of algorithms. Noise typically contributes little to the overall variance, it tends to be distributed among the lower-order principal components. By retaining only, the components that explain the majority of the variance, PCA effectively filters out the noise, leading to

a cleaner representation of the data. The noise reduction improves the performance and the generalization ability of algorithms, allowing for more accurate predictions.

However, PCA has several limitations. One of the main drawbacks is low interpretability as the resulting principal components are linear combinations of the original variables thus lacking an intuitive meaning. Furthermore, PCA assumes that the data follows a linear structure thus may not be suitable for datasets with non-linear relationships.

3.1.2.2 Removing redundant features

The selection of variables with Pearson correlation coefficient below a certain threshold is a widely used technique of feature selection in various scientific fields, although the specific threshold varies. The most popular one is the aforementioned $|r| < 0.7$, but more restrictive (e.g., 0.4 in Suzuki et al. 2008) and less restrictive (0.85 in Elith et al. 2006) thresholds have been used. Dormann et al. (2013) in their analysis confirmed, that the rule-of-thumb not to use variables correlated at $|r| > 0.7$ is a simple and effective method of variable selection.

Despite those advantages, Pearson's r is quite sensitive to non-normality (Kowalski, 1972) of the data. Because of that, a Kendall's τ correlation coefficient is proposed in this study. The filtering threshold equal to 0.7 was preserved because both Pearson's r and Kendall's τ share scale from -1 to 1.

3.2 Machine Learning Algorithms

3.2.1 Logistic Regression

Logistic Regression is a parametric model describing the relationship between a binary outcome variable and predictor variables, which can be either categorical or continuous (Hosmer et al., 2013). The outcome of the model is the estimated probability of the occurrence of an outcome given a set of predictor variables.

The logistic regression equation can be expressed as:

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p)}} \quad (2)$$

where $\mathrm{P(Y = 1|X)}$ is the probability of outcome $Y = 1$ given the set of predictors $X_1, X_2, \ldots, X_p$where $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficient of predictor variables.

Ordered Logistic Regression is an extension of Logistic Regression for ordinal dependent variable (McCullagh, 1980). One of assumptions of that model is the proportional odds assumptions, which is problematic to achieve while working with high cardinality data, so another extension of Logistic Regression is considered.

In this paper's context of multi-class classification, an extension of Logistic Regression - Multinomial Logistic Regression (MLR) is used. MLR generalized logistic regression to accommodate more than two classes and is particularly useful for cases where the outcome variable has three or more categories. MLR assumes that collinearity is relatively low, as it becomes difficult to differentiate between the impact of several variables otherwise (Goldstein, 1993).

The MLR equation can be expressed as:

$$P(Y = k|X) = \frac{e^{(\beta_{k0}+\beta_{k1}X_1+\beta_{k2}X_2+\cdots+\beta_{kp}X_p)}}{\sum_{j=1}^{K} e^{(\beta_{j0}+\beta_{j1}X_1+\beta_{j2}X_2+\cdots+\beta_{jp}X_p)}} \quad (3)$$

where $P(Y = k \mid X)$ is the probability of outcome $Y = k$ given the predictor variables $X_1, X_2, \ldots, X_p$here $\beta_{k0}, \beta_{k1}, \ldots, \beta_{kp}$ are the coefficients associated with i-th explanatory variable and the k-th outcome. An important advantage of logistic regression and its multi-class extension is that the interpretable Marginal Effects can be easily calculated, which are useful when assessing the direction and strength of predictor variables.

Performance of the MLR can be tuned mostly through regularization, a method to fight overfitting in by adding a penalty term to the loss function with associated parameter λ, indicates regularization strength (Friedman et al., 2010). Depending on the penalty term, model becomes Lasso Regression (L1 penalty) (Tibshirani 1996), Ridge Regression (L2 penalty) (Hoerl and Kennard 1970) or Elastic Net (mix of both L1 and L2 penalties) (Zou and Hastie, 2005).

During the model selection phase, MLR will not be regularized. Penalty terms will be introduced in the hyperparameter tuning process.

3.2.2 k-Nearest Neighbors

The k-Nearest Neighbors algorithm is a non-parametric model developed in 1951 (Fix and Hodges, 1951) and later expanded to the form it is known today (Cover and Hart, 1967). k-NN is also a lazy algorithm, indicating that it does not make any assumptions about the underlying data distribution and does not perform any training during the model fitting stage. Instead, it

uses the entire training dataset as its model, and makes predictions based on the similarity between new instances and the instances from the training dataset. It can be used in both regression and classification problems.

The algorithm identifies k nearest data points for each new unobserved data point and assigns a class to it based on the majority vote. The choice of k is crucial, as a value too small may result in overfitting because it will fit too closely to the training data and a value too large may result in underfitting.The proximity of instances is calculated through one of distance metrics. Common distance metrics include Euclidean, Manhattan or Minkowski distance. This paper will take into the default settings of *scikit-learn*'s KNeighborsClassifier, so the distance metric is the Minkowski distance defined as:

$$d_p(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{4}$$

with $p = 2$, so $d_p(x,y)$ becomes the Euclidean distance:

$$d(x.y) = \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

In the standard k-NN classifier, all data points are given equal weight. However, in a specific scenario, known as weighted k-NN, data points can be assigned weights based on their distance (Dudani, 1976). k-NN has several advantages, such as simplicity, flexibility, and the ability to handle non-linear relationships. However, it is sensitive to the choice of k and the distance metric, as well as its performance may worsen in high-dimensional data spaces due to the curse of dimensionality (Beyer et al., 1999).

During the model selection phase, k-NN will have default values for following four parameters:

$$\begin{cases} k & = 5 \\ p & = 2 \\ metric & = 'minkowski' \\ weights & = 'uniform' \end{cases} \tag{6}$$

3.2.3 Random Forest

Random Forest is an ensemble learning algorithm for both classification and regression tasks that combines multiple decision trees to improve the overall performance and reduce overfitting, a common issue with single decision trees (Breiman, 2001). The basic idea behind

Random Forest is to generate a large number of decision trees, each trained on a randomly selected subset of the training data. The predictions of the individual trees are combined using a majority vote (in classification) or an average (for regression) to produce the final prediction.

The number of trees $n\_estimators$ is not the only factor that can influence the performance of the Random Forest Classifier. Other parameters considered apply to individual Decision Trees (Pedregosa et al., 2011). The max_depth controls the maximum depth of each tree, as deep tree can capture more patterns in the data, but also lead to overfitting.

The next three parameters affect the split in each tree. The $min\_samples\_split$ parameter specifies the minimum number of samples required to split an internal node, $min\_samples\_leaf$ parameter determines the minimum number of samples required at a leaf node and $max\_features$ parameter determines the maximum number of features considered when looking for the best split (Pedregosa et al., 2011).

When determining the optimal split in each node, the $criterion$ parameter defines the quality measure used to evaluate it: Gini index and Entropy (Pedregosa et al., 2011).

Assuming N is the total number of classes and $p(i)$ is the probability of picking an observation of class $i \in \{1, \dots, N\}$ from set T, then Gini index is defined as (6) and measures the probability of incorrectly classifying a randomly chosen element belonging to the class from the set T (Xia et al., 2008).

$$G(T) = \sum_{i}^{N} p(i) * \big(1 - p(i)\big) \tag{7}$$

Maintaining the same assumption, Entropy is defined as (7) and is a measure of impurity or disorder in a dataset, first formulated by Shannon (1948).

$$H(T) = -\sum_{i}^{N} p(i) * log_2\, p(i) \tag{8}$$

Random Forest classification's non-parametric nature is one of its most important advantages. In contrast to parametric methods, which assume a specific form for the underlying distribution of the data, Random Forest classification makes no such assumptions. Instead, it relies on a combination of decision trees, each of which is trained on a random subset of the input features and data. This allows the algorithm to be more flexible and robust in handling complex and diverse datasets, including high-dimensional data.

Another advantage is RF's ability to handle high-dimensional and noisy data. Random Forest classification can handle noisy data more effectively than other algorithms. When dealing with noisy data, decision trees in the ensemble may produce incorrect predictions due to the presence of outliers or other sources of noise in the input data. However, since Random Forest classification is based on a combination of multiple decision trees, the overall impact of noisy or incorrect predictions is reduced, improving the overall performance of the model.

Despite its numerous advantages, Random Forest classification also has some limitations. One of the main drawbacks of Random Forest is its limited interpretability. Since it is an ensemble method composed of multiple decision trees, it can be challenging to understand how the model arrives at its final predictions. However, feature importance scores can provide some insights into which features are the most important in making those predictions. Another limitation is that the computational complexity of Random Forest can increase significantly when dealing with large datasets and complex models. This can lead to longer training and prediction times, which can be a problem in real-time or near real-time applications.

During the model selection phase, RFC will have default values for following six parameters:

$$\begin{cases} \boldsymbol{n\_estimators} & \boldsymbol{= 100} \\ \boldsymbol{max\_depth} & \boldsymbol{= None} \\ \boldsymbol{min\_samples\_split} & \boldsymbol{= 2} \\ \boldsymbol{min\_samples\_leaf} & \boldsymbol{= 1} \\ \boldsymbol{max\_features} & \boldsymbol{= 'sqrt'} \\ \boldsymbol{criterion} & \boldsymbol{= 'Gini'} \end{cases} \quad (9)$$

3.2.4 XGBoost

XGBoost (eXtreme Gradient Boosting) is an advanced implementation of Gradient Boosting Machines, a powerful and widely used ensemble learning technique for both classification and regression tasks (Chen & Guestrin, 2016).

The algorithm works by iteratively building an ensemble of weak learners (decision trees) that successively correct the errors of their predecessors. At each iteration, a new weak learner is fitted to the loss function with respect to the current learner prediction. The final prediction is obtained by combining the predictions of all weak learners using a vote. In the case of multi-class classification, raw scores are converted into class probabilities using the softmax function defined as:

$$P(y = k|\, X) = \frac{e^{\hat{f}k(x)}}{\sum_{j=1}^{K} e^{\hat{f}_j(x)}} \tag{10}$$

here $P(y = k|x)$ is the predicted probability of class k for observation x and $\hat{f}k(x)$ is the raw score for class k and K is the total number of classes.

The performance of XGBoost classification can be optimized by adjusting various parameters. Chen and Guestrin (2016) classified these parameters into three categories: general, booster, and learning task parameters. The first type relates to the type of booster, the second – to the booster itself and the third – to the learning task and its objective.

By default, the *booster* parameter in XGBoost utilizes tree models (Chen and Guestrin, 2016), inheriting certain parameters from decision trees as described in Section 3.2.3.: *n_estimators* and *max_depth*. Additional parameters are related to the boosting process. During each boosting round, the *learning_rate* parameter regulates its step size, the *subsample* parameter determines the fraction of training samples that are randomly selected and used for training and the *gamma* parameter sets the minimum loss required to initiate another split in a tree, thereby controlling the complexity of a tree. XGBoost introduces several improvements to the standard gradient boosting framework, such as regularized learning, which helps prevent overfitting, and efficient tree construction algorithms that enable faster and more accurate tree learning (Chen & Guestrin, 2016). Moreover, XGBoost supports parallel and distributed computing, making it highly scalable and suitable for large datasets.

XGBoost seems similar to Random Forest, however the primary difference between them lies in their learning strategy. RF employs bootstrap aggregating (bagging), wherein multiple decision trees are built independently and in parallel, and their predictions are aggregated through average (in regression) or vote (in classification) to obtain the final prediction. XGB uses boosting, a sequential learning technique where trees are built iteratively, with each new tree focusing on correcting the eros made by the previous tree.

During the model selection phase, XGB will have default values for following six parameters:

$$\begin{cases} booster & = gbtree \\ n\_estimators & = 100 \\ max\_depth & = 10 \\ learning\_rate & = 0.1 \\ subsample & = 1 \\ gamma & = 0 \end{cases} \tag{11}$$

3.3 Model selection and evaluation

Models are selected based on their statistical metrics and then evaluated using performance metrics based on Kosc et al. (2019), Bui and Ślepaczuk (2021) and Magdon-Ismail et al. (2004).

3.3.1 Statistical metrics

3.3.1.1 Balanced Accuracy

Balanced Accuracy is a good metric for evaluating classification models in imbalanced problems because it considers the distribution of the classes. In imbalanced problems, where the number of instances in one class is significantly larger or smaller than the others, traditional accuracy metrics can be misleading. For example, a classifier that always predicts the majority class can have a high accuracy, even though it does not perform well on the minority class. Balanced Accuracy for binary problem is defined as the arithmetic average of sensitivity and recall:

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{12}$$

here is no agreed-upon definition for extending Balanced Accuracy to multi-class classification problems, but this study will use the definition proposed by Mosley (2013), Guyon et al. (2015), and Kelleher (2015), as the Recall of each class averaged by the number of classes:

$$Balanced\ Accuracy = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k}{TP_k + FN_k} \tag{13}$$

this approach provides a measure of the overall performance of a classifier that is less sensitive to imbalanced class distributions in multi-class classification problems.

3.3.1.2 F1-score

The F1 score is a metric that combines both precision and recall into a single value to evaluate the overall performance of a classifier. It is particularly useful in imbalanced problems where the distribution of classes is skewed, as it considers both false positives and false negatives. The F1 score is the harmonic mean of precision and recall, and is defined as:

$$F1 - Score = 2 \times \frac{\frac{TP}{TP + FP} \times \frac{TP}{TP + FN}}{\frac{TP}{TP + FP} + \frac{TP}{TP + FN}} \tag{14}$$

In imbalanced problems, where the number of instances in one class is significantly larger or smaller than the others, traditional accuracy metrics can be misleading. A classifier that predicts the majority class may have a high accuracy but may not perform well on the minority class. The F1 score provides a more accurate measure of the overall performance of a classifier in such cases, by balancing the trade-off between precision and recall.

In multiclass classification problems, the F1-score is calculated by finding the weighted average of the metrics for each label, where the weight is Support score. This approach, known as weighted averaging, considers the label imbalance, which can result in an F1-score that is not between precision and recall, unlike in binary classification problems. This approach provides a more accurate measure of the overall performance of a classifier in multiclass classification problems, particularly in cases where the class distribution is imbalanced.

3.3.1.3 Hyperparameter tuning

Hyperparameter tuning is a process of optimizing the hyperparameters (set before training the algorithm) of a Machine Learning algorithm to improve its performance. Literature (Wu et al., 2019; Bergstra and Bengio, 2012) distinguishes traditional and advanced types of this process.

Grid Search is a one of traditional ways of performing hyperparameter optimization and relies on exhaustive searching through the whole hyperparameter space of an algorithm. This method entails selecting a finite set of values for each hyperparameter and training the model using all possible combinations. Trying all possible combinations is also a limitation of this approach, as Grid search gets more computationally expensive with increasing parameter space or with spare data.

Random Search uses only a subset of hyperparameters from the search space and train the model using these hyperparameters. This approach avoids the computational burden of a Grid search and can be more efficient in finding best parameter configurations, especially when the parameter space is large, or data is sparse. Although this approach is less computational expensive than Grid search, it also has limitation – it does not guarantee the optimal set of parameters and may require many iterations to find a sufficient set of parameters.

Snoek et al. (2012) highlights limitations of those two traditional approaches and propose the Bayesian optimization as a better hyperparameter tuning technique. Bayesian optimization is a statistical approach that constructs a probabilistic model of the objective function, which maps hyperparameter values to the validation set's objective value. It iteratively evaluates promising hyperparameter configurations based on the current model and updates it to acquire

observations revealing as much information as possible about this function, particularly the location of the optimum (Wu et al., 2019). It attempts to balance exploration of hyperparameters with uncertain outcomes and exploitation of those hyperparameters expected to be near the optimum. Empirical evidence (Snoek et al., 2012) has demonstrated that Bayesian optimization outperforms grid search and Random Search in terms of fewer evaluations required to achieve better results, mainly due to the ability to reason about the quality of experiments before conducting them.

3.3.2 Performance metrics

3.3.2.1 Annualized Return Compounded (ARC)

The Annualized Return Compounded (ARC) is a financial metric that is widely used by investors to evaluate the performance of an investment over a certain period and compare the returns of different investments and evaluate the effectiveness of their investment strategies (Kosc et al., 2019). It represents the compounded rate of return that an investment has generated over a year. The ARC is expressed as a percentage (%) and is calculated by taking the product of the growth rate of each period over the sample size and then subtracting one from the result.

The formula for ARC is represented as:

$$ARC = \prod_{i=1}^{N}(1 + R_i)^{252/N} - 1 \tag{15}$$

where R_i is the percentage rate of return and N is the sample size.

3.3.2.2 Annualized Standard Deviation (ASD)

The Annualized Standard Deviation (ASD) is a financial metric that is used to measure the risk associated with an investment and assess its volatility (Kosc et al., 2019). The ASD is expressed as a percentage (%) and is calculated by taking the square root of the product of the sample size and the variance of the returns.

The formula for ASD is represented as:

$$ASD = \sqrt{252} * \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(R_i - \bar{R})^2} \tag{16}$$

where R_i is the percentage rate of return, $\bar{R}$ is the average rate of return and N is the sample size.

3.3.2.3 Maximum Drawdown (MDD)

Maximum Drawdown (MDD) is a financial metric that measures the maximum loss suffered by an investor during a certain period (Magdon-Ismail et al., 2004). It is calculated as the difference between the global maximum and the consecutive global minimum of the equity curve expressed as a percentage.

$$MDD(N) = \sup_{t\in[0,N]}\left[\sup_{s\in[0,t]}\frac{X(s)-X(t)}{X(s)}\right] \tag{17}$$

Where $X(n)$ is the price process in the time point n and N is the sample size.

3.3.2.4 Equity curve

The equity provides a graphical depiction of the performance of an investment strategy within a specified timeframe. The curve tracks the changes in investment value over time, enabling a comprehensive overview of the strategy's overall performance. The visual representation of the equity curve provides an intuitive view that can help identify trends, patterns, and strong or weak performance periods (Kisela et al., 2015). Comparing the equity curves of multiple investment strategies on a shared graph provides an efficient method to identify the most effective strategy. The side-by-side plotting of equity curves enables for an easy comparison the performance of different strategies, recognition of their strengths and weaknesses.

4 Results

4.1 Model selection

The model space for each metal encompassed two feature selection methods (Kendall-tau filtering and PCA), five different frequencies (1, 5, 10, 15 and 20 trading days) and four algorithms (MLR, k-NN, RFC and XGB), resulting in a total of 40 possible combinations. Each combination was evaluated using three metrics: Balanced Accuracy (BA), F1-score and Annualized Return Compounded (ARC), calculated on the training data (2000-2018) using Time Series cross-validation methods as outlined in Section 3.1. The best-performing model based on each individual metric was subsequently selected for hyperparameter tuning.

Figure 5. showcases the optimal combinations for predicting Gold price movements, following the outlined methodology. Despite evaluating each combination on three distinct statistics, only two models were selected, because the Random Forest Classifier trained on the 15 days frequency price movement data using Kendall-tau filtering as a feature selection performed best in terms of Balanced Accuracy: **0.3957** and Annualized Return Compounded: **0.1046**, meaning that strategy brought on average **10.46%** annually in the period 2000-2018. The k-Nearest Neighbors algorithm trained on 20 days frequency price movement data using PCA technique, was the best model in terms of F1-score, achieving **0.3807**. This model also has a lower value of Drawdown: **8.95%**, compared to the **10.27%** in RFC.

Figure 5. Best performing models for Gold (XAU/USD) market direction prediction

Metal	Frequency	Feature Selection	Model	Train				
				BA	F1	ARC	DD	ASD
Gold	15D	Kendall-tau	RFC	**0.3957**	0.3517	**0.1046**	0.1027	0.0016
Gold	20D	PCA	KNN	0.3839	**0.3807**	0.0472	0.0895	0.0011

Note: Metal – one of three metals, Frequency – time frequency of the data in trading days (D – trading day), Feature Selection – one of the feature selection methods: Kendall-tau filtering or Principal Component Analysis, Model – one of four models: MLR, KNN, RFC or XGB, Train – indication that statistics are calculated on the training subset (2000-2018), BA – Balanced Accuracy, F1 – F1-score, ARC – Annualized Return Compounded, DD – Drawdown, ASD – Annualized Standard Deviation.

Source: Own calculations

Figure 6. illustrates the optimal combinations for predicting Silver price movements. In contrast to Gold (Figure 5.), there are now three top-performing combinations, with each excelling in a different evaluation metric. The k-Nearest Neighbors model, trained on daily data and utilizing Kendall-tau filtering as a feature selection method, achieved the highest F1-score: **0.3491**. In terms of Annualized Return Compounded, the eXtreme Gradient Boosted model,

trained on daily data and employing PCA for dimensionality reduction, outperformed the other models with an impressive value of **23.28%**. That XGB model achieved similar values of BA and F1-score to the k-NN, however the achieved ARC is over 20 times higher, meaning that strategy brought on average 23.28% return annually! The best performing model in terms of BA turned out to be XGB with PCA trained on the 20 days frequency price movement data, achieving **0.4049**.

Figure 6. Best performing models for Silver (XAG/USD) market direction prediction

Metal	Frequency	Feature Selection	Model	Train				
				BA	F1	ARC	DD	ASD
Silver	1D	Kendall-tau	KNN	0.3464	**0.3491**	0.0124	0.3141	0.0047
Silver	1D	PCA	XGB	0.3494	0.3429	**0.2328**	0.1810	0.0053
Silver	20D	PCA	XGB	**0.4049**	0.3448	0.0643	0.2002	0.0045

Note: Metal – one of three metals, Frequency – time frequency of the data in trading days (D – trading day), Feature Selection – one of the feature selection methods: Kendall-tau filtering or Principal Component Analysis, Model – one of four models: MLR, KNN, RFC or XGB, Train – indication that statistics are calculated on the training subset (2000-2018), BA – Balanced Accuracy, F1 – F1-score, ARC – Annualized Return Compounded, DD – Drawdown, ASD – Annualized Standard Deviation.

Source: Own calculations

Figure 7. demonstrates the best combinations for predicting Platinum price movements. Only two models were selected, achieving similar values of all three metrics. The best model in terms of BA and F1-score turned out to be RFC model with PCA technique trained on daily data, achieving respectively: **0.3719** and **0.3621**, while XGB with PCA technique trained on daily data as well, achieved the highest score of ARC: **15.63%**, a value comparable with ARC achieved by the other model (**12.90%**), however the Drawdown was significantly lower: **23%**, compared to **32.70%** achieved by the RFC.

Figure 7. Best performing models for Platinum (XPT/USD) market direction prediction

Metal	Frequency	Feature Selection	Model	Train				
				BA	F1	ARC	DD	ASD
Platinum	1D	PCA	RFC	**0.371900**	**0.362100**	0.129000	0.327000	0.002900
Platinum	1D	PCA	XGB	0.363800	0.360300	**0.156300**	0.230000	0.002900

Note: Metal – one of three metals, Frequency – time frequency of the data in trading days (D – trading day), Feature Selection – one of the feature selection methods: Kendall-tau filtering or Principal Component Analysis, Model – one of four models: MLR, KNN, RFC or XGB, Train – indication that statistics are calculated on the training subset (2000-2018), BA – Balanced Accuracy, F1 – F1-score, ARC – Annualized Return Compounded, DD – Drawdown, ASD – Annualized Standard Deviation.

Source: Own calculations

It is important to note that the choice of model, feature selection method and data frequency can have a significant impact on performance metrics, encompassing both statistical (BA, F1-

score) and financial indicators (ARC, DD, ASD). No universal solution exists for this problem, as the most suitable combinations vary depending on the specific task, whether the goal is to maximize BA, ARC or other relevant statistics. An interesting observation across all three noble metals is the absence of Multinomial Logistic Regression (MLR) as a selected algorithm in any case. This outcome was somewhat anticipated, as MLR is a parametric model that possesses limited capability to capture the underlying relationships between variables when compared to tree-based models or k-Nearest Neighbors (k-NN).

Another noteworthy discovery applicable to all three precious metals is that investment strategies constructed on models trained solely using Technical Analysis (TA) indicators data managed to attain positive ARC, ranging from **1.24%** to **23.28%**, depending on the specific metal. Additionally, investment strategies for Silver and Platinum exhibited a higher degree of Drawdown, varying from **18.1%** to **32.7%**, in contrast to Gold, which demonstrated a lower-level ranging from **4.72%** to **10.46%**. This observation aligns with a higher volatility observed in the daily returns of Silver and Platinum, as illustrated in Figure 3.

Additional observation is that there appears to be no discernible correlation between statistical measures and the financial profitability of the strategy, as indicated by the ARC statistic. Specifically, when examining the data for Gold, it was found that the model with the highest ARC also had the highest BA. However, this relationship did not hold true for Silver or Platinum. Furthermore, the absence of a consistent relationship between BA, F1-score and ARC becomes even more apparent when analyzing strategies built on Silver and Platinum data. Despite minimal discrepancies in BA and F1-scores across different models, the ARC exhibited significant variations ranging from approximately **1%** to **23%**.

4.2 Model evaluation

4.2.1 Hyperparameter tuning

The model selection process involved identifying, for each metal (Gold, Silver, and Platinum), a combination of model type (MLR, k-NN, RFC, XGB), data frequency (1, 5, 10, 15, 20 trading days), and variable selection method (Kendall-tau filtering, PCA) that yielded the highest possible scores for metrics: BA, F1-score and ARC. For each metal, 2 (for Gold and Platinum) or 3 (for Silver) top-performing combinations, which were then subjected to a hyperparameter tuning process.

The hyperparameter tuning process is conducted using the Randomized Search method, which unlike Grid Search, does not test all possible combinations from the declared hyperparameter space. Instead, it randomly selects a specified number (in this case, $N = 50$) of combinations and chooses the best one among them. To ensure results reproducibility, a seed value is specified by the $random\_state$ parameter. Randomized Search significantly reduces computational costs, however, there is a risk that the obtained combination of hyperparameters may not be optimal.

Among the selected combinations, none of them included the MLR model. Therefore, when describe the hyperparameter space for each model, emphasis will be placed on the remaining three models: k-NN, RFC and XGB.

The hyperparameter space, denoted as HP for each model is defined as follows (default values of those hyperparameters were described in Section 3.2.1 – 3.2.4):

$$HP = \begin{cases} HP_{kNN} = \begin{cases} k & \in \{1, 2, \dots, 10\} \\ p & \in \{1, 2\} \\ metric & \in \{minkowski, manhattan, euclidean\} \\ weights & \in \{uniform, distance\} \end{cases} \\ HP_{RFC} = \begin{cases} n\_estimators & \in \{100, 110, \dots, 150\} \\ max\_depth & \in \{None, 5, 10, \dots, 30\} \\ min\_samples\_split & \in \{2, 5, 10\} \\ min\_samples\_leaf & \in \{1, 2, 4\} \\ max\_features & \in \{None, sqrt, log\ 2\} \end{cases} \\ HP_{XGB} = \begin{cases} booster & \in \{gbtree\} \\ n\_estimators & \in \{100, 110, \dots, 150\} \\ max\_depth & \in \{None, 1, 2, \dots, 10\} \\ learning\_rate & \in \{0.01, 0.1, 0.5\} \\ gamma & \in \{0, 0.1, \dots, 0.4\} \end{cases} \end{cases} \tag{18}$$

The models are evaluated using a separate test subset of the data that covers the period from 2018 to 2022. During the hyperparameter tuning process using Randomized Search, different combinations of hyperparameters are tested and evaluated on the training subset of the data, using Time Series cross-validation technique outlined in the Section 3.1. However, it is important to outline once again that even though Randomized Search aims to find the best hyperparameters combinations, there is no guarantee that the obtained combination will be optimal. In some cases, the combination found through Randomized Search may even perform

worse than the model with default parameter values. In such cases, Figure 8. – Figure 10. will include models with their default parameter values.

The column *Scorer* specifies the metric against which the hyperparameters were tuned. This metric represents the performance measure used to evaluate and compare the models. It is a metric in which the baseline model, with its default parameter values, outperformed the other models.

Ultimately, from the 2-3 models that underwent hyperparameter tuning process, the best model is selected based on its highest ARC value on the test subset.

The case of Gold (Figure 8.) illustrates that model trained on the 15 trading days data frequency yielded a hyperparameter combination that performed worse than the baseline model. However, the model trained on data with a frequency of 20 trading days showed improved F1 metric value compared to the baseline model (Figure 5.). That metric increased from **0.3807** to **0.3991** on the training data, accompanied by a slight increase in Balanced Accuracy from **0.3839** to **0.3871**. However, the ARC metric decreased by over 1 percentage point from **4.71%** to **3.33%**.

Although the k-NN model trained on 20D data frequency achieved higher BA and F1 metric values on the test subset, the RFC model trained on 15D data frequency with Kendall-tau filtering as a feature selection method achieved higher ARC metric values on both train and test subsets and thus was selected as the best model, respectively: **10.46 %** and **8.57%**. The model with BA and ARC as *Scorer* metrics was the model which produced the best trading signals. A relationship true for Gold but it will not hold true in the remaining two cases.

Figure 8. Selected models for XAU/USD (Gold) after hyperparameter tuning

Metal	Fre-quency	Feature Selection	Model	Scorer	Data Subset	BA	F1	ARC	DD	ASD
Gold	15D	Kendall-tau	RFC	BA & ARC	Train	**0.3957**	0.3517	**0.1046**	0.1027	0.0016
					Test	0.3271	0.3120	**0.0857**	0.1898	0.0008
Gold	20D	PCA	KNN	F1	Train	0.3871	**0.3991**	0.0333	0.1348	0.0013
					Test	**0.3932**	**0.3785**	-0.0033	0.1892	0.0002

Note: Metal – one of three metals, Frequency – time frequency of the data in trading days (D – trading day), Feature Selection – one of the feature selection methods: Kendall-tau filtering or Principal Component Analysis, Model – one of four models: MLR, KNN, RFC or XGB, Train – indication that statistics are calculated on the training subset (2000-2018), BA – Balanced Accuracy, F1 – F1-score, ARC – Annualized Return Compounded, DD – Drawdown, ASD – Annualized Standard Deviation.

Source: Own calculations

In the case of Silver (Figure 9.), the tuning process improved improves the F1 score for the k-NN model from **0.3491** to **0.3685** and the BA score for the XGB model trained on the 20D frequency data from **0.4049** to **0.4183** on the training subset, in comparison to baseline models (Figure 6.)

Nevertheless, the model that achieved the best results in terms of both BA (**0.3759**) and F1 (**0.3766**) metrics on the test dataset was the baseline XGB model trained on daily frequency data with PCA as a dimensionality reduction technique. That model was also the one with the highest value of ARC on both train (**23.28%**) and test (**4.84%**) subsets. The selected model was the one with ARC alone as the *Scorer* metric, not together with BA, as was the case for Gold (Figure 8.)

Figure 9. Selected models for XAG/USD (Silver) after hyperparameter tuning

Metal	Frequency	Feature Selection	Model	Scorer	Data Subset	BA	F1	ARC	DD	ASD
Silver	1D	Kendall-tau	KNN	F1	Train	0.3602	**0.3685**	0.0199	0.3006	0.0050
					Test	0.3454	0.3456	0.0245	0.4284	0.0011
Silver	1D	PCA	XGB	ARC	Train	0.3494	0.3429	**0.2328**	0.1810	0.0053
					Test	**0.3759**	**0.3766**	**0.0484**	0.2694	0.0041
Silver	20D	PCA	XGB	BA	Train	**0.4183**	0.3362	0.0518	0.1150	0.0046
					Test	0.2940	0.2455	-0.0421	0.5975	0.0010

Note: Metal – one of three metals, Frequency – time frequency of the data in trading days (D – trading day), Feature Selection – one of the feature selection methods: Kendall-tau filtering or Principal Component Analysis, Model – one of four models: MLR, KNN, RFC or XGB, Train – indication that statistics are calculated on the training subset (2000-2018), BA – Balanced Accuracy, F1 – F1-score, ARC – Annualized Return Compounded, DD – Drawdown, ASD – Annualized Standard Deviation.

Source: Own calculations

The example of Platinum (Figure 10.) clearly highlights the main drawback of Randomized Search, which is that the obtained hyperparameter combination may not be optimal. This was the case for both XPT/USD models, because of which Figure 10. Displays the baseline models for both scenarios.

It is worth noting that the model which showed the highest ARC metric value on the training subset (**15.63%**) achieved a significantly worse result of nearly 30 percentage points lower on the test subset (**-14.29%**)!

The model that achieved the best ARC score on the test subset was the RFC model trained on daily data with PCA as a dimensionality reduction technique and BA and F1 as the *Scorer* metrics, unlike Silver (which had ARC) and partially like Gold (which had BA and ARC).

Figure 10. Selected models for XPT/USD (Platinum) after hyperparameter tuning

Metal	Frequency	Feature Selection	Model	Scorer	Data Subset	BA	F1	ARC	DD	ASD
Platinum	1D	PCA	RFC	BA & F1	Train	**0.3719**	**0.3621**	0.1290	0.3270	0.0029
					Test	**0.3691**	**0.3765**	**0.0944**	0.2898	0.0042
Platinum	1D	PCA	XGB	ARC	Train	0.3638	0.3603	**0.1563**	0.2300	0.0029
					Test	0.3361	0.3442	-0.1429	0.6945	0.0041

Note: Metal – one of three metals, Frequency – time frequency of the data in trading days (D – trading day), Feature Selection – one of the feature selection methods: Kendall-tau filtering or Principal Component Analysis, Model – one of four models: MLR, KNN, RFC or XGB, Train – indication that statistics are calculated on the training subset (2000-2018), BA – Balanced Accuracy, F1 – F1-score, ARC – Annualized Return Compounded, DD – Drawdown, ASD – Annualized Standard Deviation.

Source: Own calculations

The hyperparameter tuning process did not yield significant improvements to the overall performance. That can be attributed to the limitations of the Randomized Search approach, which does not guarantee the globally optimal sets of hyperparameters. The relatively small computational complexity of Randomized Search could be overshadowed by its inherent pseudo-randomness (or even pure randomness, if $random\_state$ is not declared), which may result in high variance in performance evaluation, leading to difficulty in identifying the best performing hyperparameters.

4.3 Equity curves

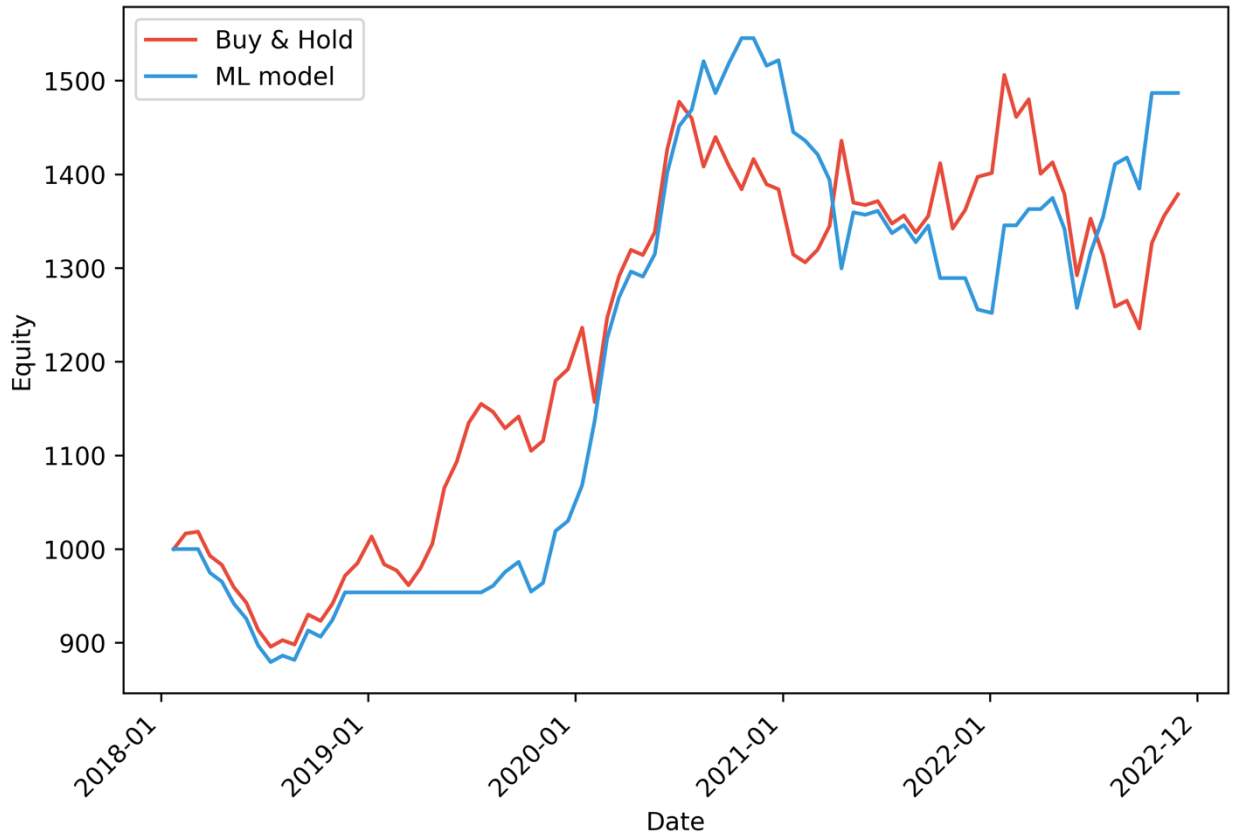
The equity curves presented for each noble metal illustrate the investment strategy built on the signals derived from the best performing models. The strategy employs a set of basic rules, where the initial balance is set to 1,000 USD, and a long or short position is opened for every BUY or SELL signal, respectively, with the entire account at the opening price of the subsequent candle. The position is closed at the closing price of the same candle. To provide a baseline for comparison, a simple Buy & Hold strategy is also plotted.

4.3.1 Gold

The equity curves for Gold (Figure 11.) reveal that the investment strategy based on the baseline Random Forest model trained on the 15D frequency with Kendall-tau filtering, outperformed the Buy & Hold strategy during the testing period (01/2018 – 12/2022). However, for most of that period, the strategy performed worse than Buy & Hold, apart for two short periods in late 2020 and the second half of 2022. It was the latter period that contributed to the fact that in the end strategy yielded bigger return than Buy & Hold. It is worth noting that the overall Drawdown during the entire period was only **8.57%**. Although the second model, trained on

the 20D frequency data, exhibits outstanding BA and F1-score, it did not result in better performing investment strategies. Another noteworthy fact is that Random Forest Classifier was successful in outperforming the market in Gold, this may not hold true for other metals as well.

Figure 11. Equity curve of ML-based investment strategy on Gold in comparison to Buy & Hold



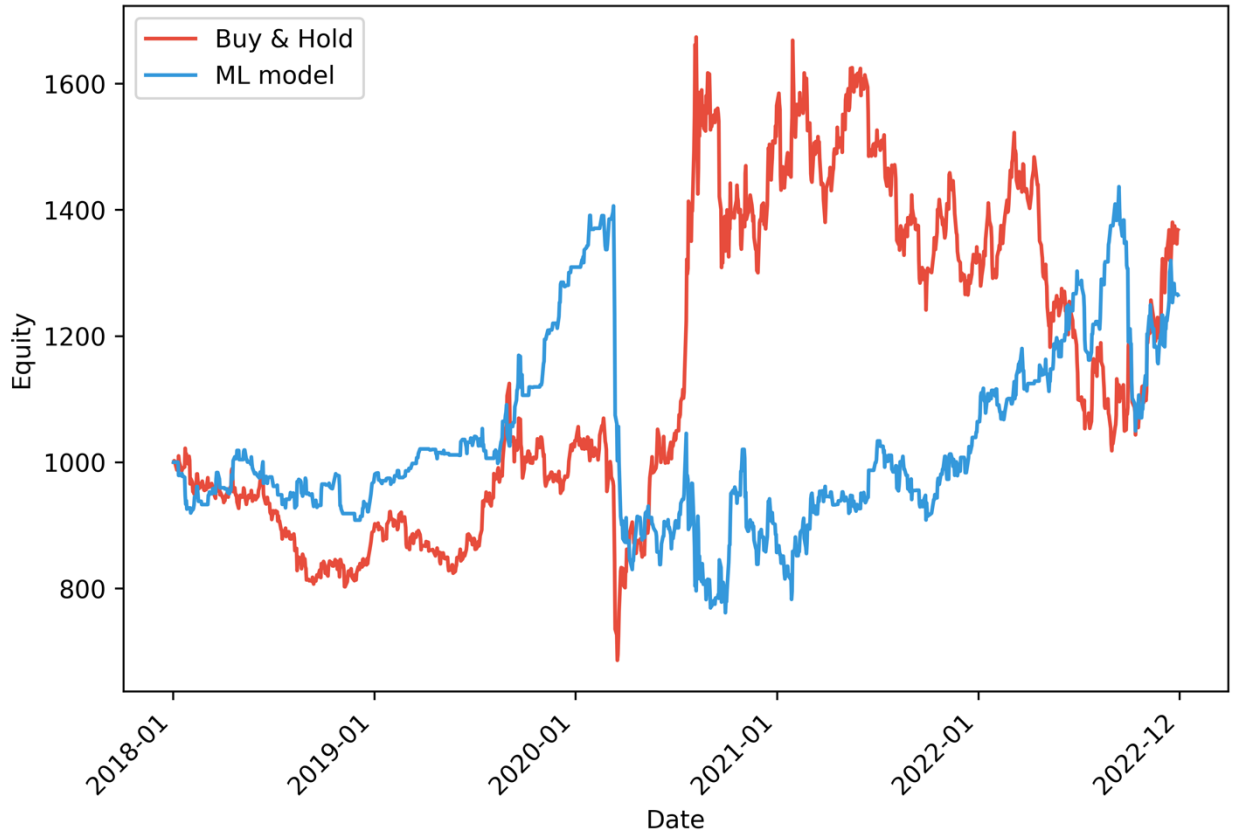
Source: Own calculations

4.3.2 Silver

The results for Silver, as depicted in Figure 12., differ from those of Gold. The equity curve illustrates the performance of the strategy based on the k-NN model, tuned against the F1-score metric, trained on daily data with Kendall-tau filtering, compared to the Buy & Hold approach during the period from 01/2018 to 12/2022. The graph shows a clear division in the strategy's performance: until the first quarter of 2020, the strategy tended to outperform the market. However, around the time of the pandemic breakout, the strategy experienced a significant loss, resulting in a Drawdown of **26.94%**. Ultimately, after the five-year period, both the ML-based strategy and Buy & Hold yielded comparable results.

In contrast to Gold, where the best model was the baseline RFC trained on 15D frequency data, in the case of Silver, the best model turned out to be k-NN trained on daily data and tuned against the F1-score metric. Up to this point, the feature selection method of Kendall-tau filtering was found to be effective for the best strategies in both Gold and Silver. However, it should be noted that this dependency will not hold true for Platinum, as for that metal the relationship between the feature selection method and the performance will differ.

Figure 12. Equity curves of ML-based investment strategy on Silver in comparison to Buy & Hold



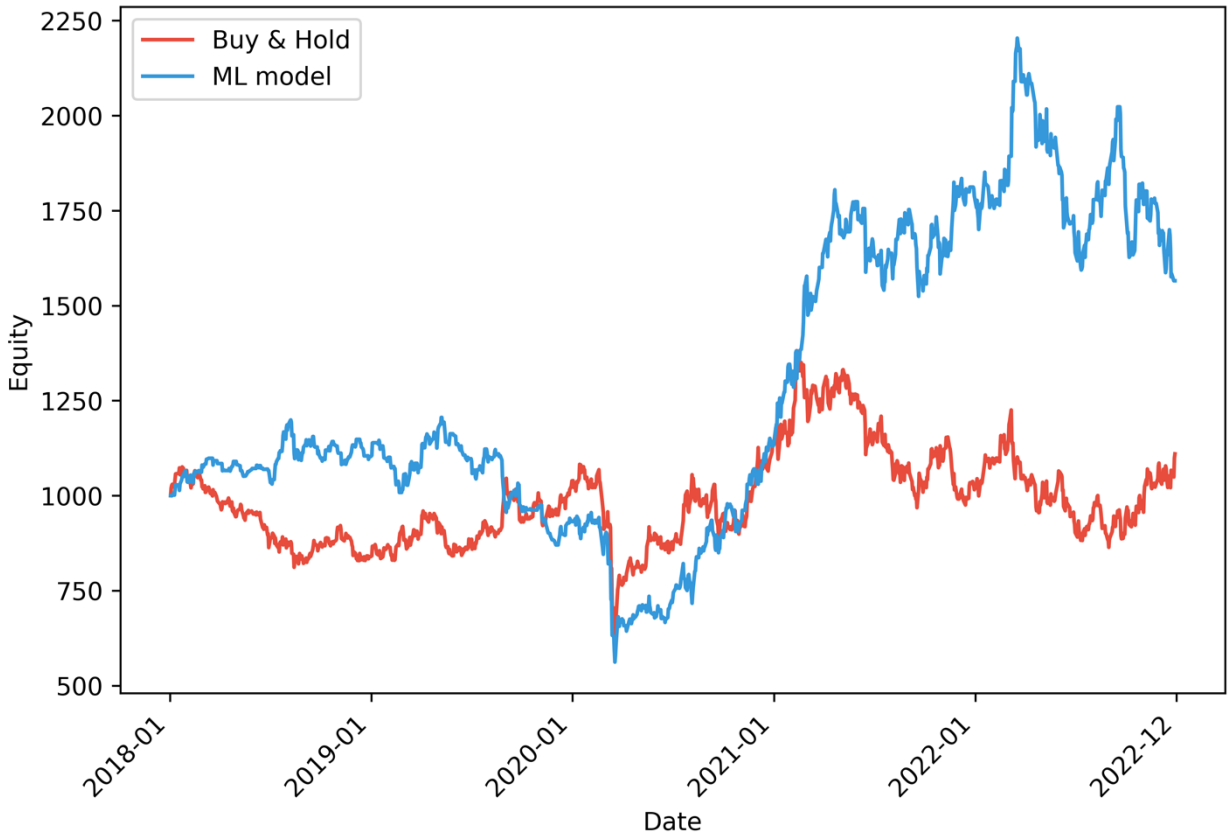
Source: Own calculations

4.3.3 Platinum

Figure 13. Illustrates the investment strategy base on the baseline RFC model, trained on daily data with PCA as a dimensionality reduction technique. The strategy performed well overall during the period from 01/2018 to 12/2022. It experienced a significant loss in the first quarter of 2020, following the outbreak of the pandemic, resulting in Drawdown of **28.98%**. However, the strategy quickly recovered from this loss, and for the remainder of the testing period, it outperformed the market to a great extent.

The best model for Platinum turned out to be the baseline RFC model (as in Gold), trained on daily data (as in Silver), using PCA as the dimensionality reduction technique (unlike in any other metal). That difference highlights the lack of a consistent relationship between specific components (data frequency, feature selection, model) and overall performance.

Figure 13. Equity curves of ML-based investment strategy on Platinum in comparison to Buy & Hold



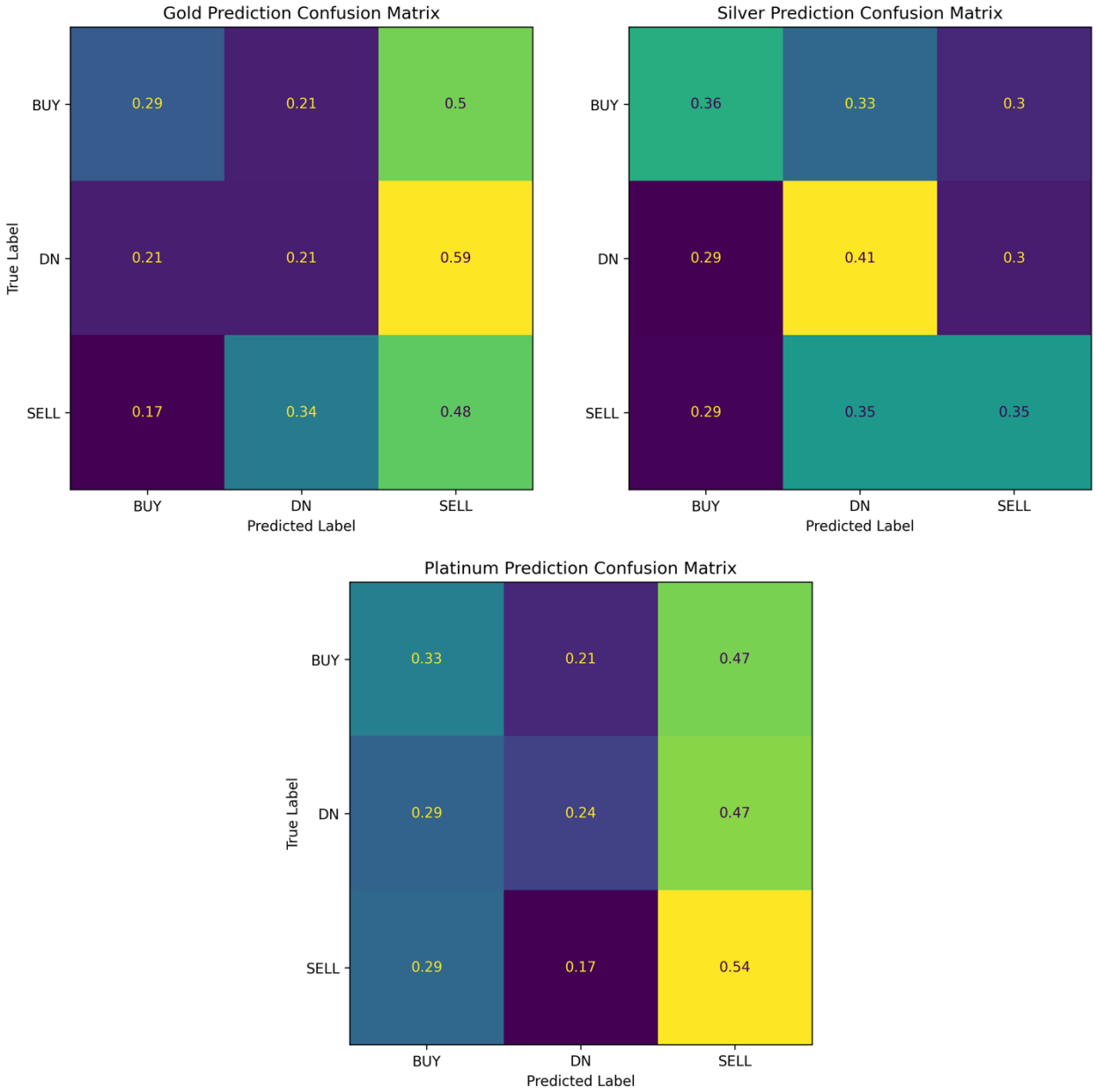
Source: Own calculations

4.4 Confusion matrices

In the context of classification models evaluation, confusion matrices serve as a useful tool for analyzing the accuracy of model predictions because it can show if and which class is over- or underpredicted by the given model.

The analysis of confusion matrices across all three precious metals (Figure 14.) indicates that in case of Gold and Platinum there tends to be an overprediction of class SELL. At the same time, those two models outperformed the market in the testing period, while in the case of Silver, where prediction of classes is more equal strategy yielded comparable (but slightly worse) results than the market.

Figure 14. Confusion Matrices for best performing models in each metal



Source: Own calculations

5 Conclusions

In this paper, the performance of various Machine Learning algorithms: Multinomial Logistic Regression, k-Nearest Neighbors, Random Forest and XGBoost in predicting the direction of the price movements of three precious metals: Gold, Silver and Platinum pairs correlated with US Dollar listed on the Foreign Exchange. Models were evaluated with a Time Series cross-validation process and then chosen based on cross-validated metrics: Balanced Accuracy and F1-score. It was found that the best performing models varied depending on the metal being analyzed and the specific evaluation metric being used. The behavior of various combinations of models, data frequency (1, 5, 10, 15, 20 trading days), and feature selection methods (Kendall-tau filtering, PCA) varies depending on the specific context. However, a common observation across all three metals is that investment strategies based on selected models were able to generate positive returns. Additionally, in two out of three cases (Gold and Platinum), these strategies even outperformed the market.

The performance of various investment strategies based on the signals generated by these models was evaluated as well. Simple Buy & Hold strategy was compared as a baseline with more complex strategies based on the Machine Learning model. It was found that past performance in the form of Balanced Accuracy, F1 and ARC scores achieved on the training data subset cannot indicate the future performance. There is no rule that could be generalized over all three noble metals. For Gold the best strategy was built on model with the highest BA & ARC score, for Silver – with the highest ARC score and for Platinum – with the highest BA & F1-score. Study suggests that a "one-size-fits-all" approach to machine learning models is not suitable for noble metal price movement prediction. Each metal requires a tailored approach for optimal performance and there is no singular model that would be the best choice in each case.

Random Forest, k-NN and XGBoost have been shown to be highly effective in predicting movements using Technical Analysis indicators only, without relying on creation of mathematical rules based on those indicators. Unlike traditional rule-based models, black-box models can capture the subtle nuances of market behavior, expressed by TA indicators. However, usage of black-box models also comes with some limitations, including their complexity and lack of interpretability, especially combined with PCA.

While the study aimed to investigate the performance of Machine Learning algorithms on noble metals price movement using Technical Analysis indicators, it is important to note the limitations of this research. One limitation was the introduction of a threshold selection method,

aiming for equal distribution across all three classes. This approach may not capture the nuances of the market and could limit the effectiveness of the models. Future studies in this field could explore the usage of more dynamic thresholds that adjust to changing market conditions.

Another limitation was the lack of transaction costs. In real-world scenarios, transaction costs can have a significant impact on the profitability of a strategy, as argue Bajgrowicz and Scaillet (2012). Incorporating transaction costs could provide more realistic evaluation of the performance of the investment strategies, especially for relatively rare traded noble metal such as Platinum.

The study focused solely on Technical Analysis indicators as predictors, without considering other variables that could potentially impact the price movements of precious metals, such as macroeconomic indicators or news events. Macroeconomic news events may be a valuable source of information for predicting the price movements, but only on a relatively high frequency such as daily. Considering news events on weekly or monthly frequency is pointless since the news does not have the effect on the market over such a long period of time. However, investigating the impact of additional variables could potentially improve the results.

Additionally, only a basic methods of feature selection were considered in this study. Fact that the problem is a multi-class classification limits the possible methods of feature selection. More nuanced methods, such as ReliefF, the multi-class extension of a Relief, introduced first by Kira and Rendell (1992), possibly outperform traditional methods, thus reducing data dimension even more.

Lastly, the hyperparameter tuning process only utilized the Random Search approach. While this method was primarily used for the purpose of reducing the computational complexity of model training, it may not find the optimal set of hyperparameters. A more nuanced approach, such as Bayesian optimization may be more beneficial, as it was proved to be more effective than standard methods such as Random or Grid Search (Snoek et al., 2012). Despite these limitations, the study provides a valuable insight into the performance of Machine Learning models in predicting price movements of noble metals, questioning once again the Efficient Market Hypothesis.

References:

Aguirre, A. A. A., Medina, R. A. R., & Méndez, N. D. D. (2020). Machine learning applied in the stock market through the Moving Average Convergence Divergence (MACD) indicator. *Investment Management & Financial Innovations*, *17*(4), 44.

Bajgrowicz, P., & Scaillet, O. (2012). Technical trading revisited: False discoveries, persistence tests, and transaction costs. *Journal of Financial Economics*, *106*(3), 473-491.

Bajgrowicz, P., & Scaillet, O. (2012). Technical trading revisited: False discoveries, persistence tests, and transaction costs. *Journal of Financial Economics*, *106*(3), 473-491.

Bampinas, G., & Panagiotidis, T. (2015). Are gold and silver a hedge against inflation? A two century perspective. *International Review of Financial Analysis*, *41*, 267-276.

Ban, G. Y., El Karoui, N., & Lim, A. E. (2018). Machine learning and portfolio optimization. *Management Science*, *64*(3), 1136-1154.

Barak, S., & Modarres, M. (2015). Developing an approach to evaluate stocks by forecasting effective features with data mining methods. *Expert Systems with Applications*, *42*(3), 1325-1339.

Baur, D. G., & Lucey, B. M. (2010). Is gold a hedge or a safe haven? An analysis of stocks, bonds and gold. *Financial review*, *45*(2), 217-229.

Baur, D. G., & McDermott, T. K. (2010). Is gold a safe haven? International evidence. *Journal of Banking & Finance*, *34*(8), 1886-1898.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).

Bernanke, B. S., Gertler, M., & Gilchrist, S. (1994). The financial accelerator and the flight to quality.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful?. In *Database Theory—ICDT'99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7* (pp. 217-235). Springer Berlin Heidelberg.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *The Journal of finance*, *47*(5), 1731-1764.

Bustos, O., & Pomares-Quimbaya, A. (2020). Stock market movement forecast: A systematic review. *Expert Systems with Applications*, *156*, 113464.

Capie, F., Mills, T. C., & Wood, G. (2005). Gold as a hedge against the dollar. *Journal of International Financial Markets, Institutions and Money*, *15*(4), 343-352.

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

Coudert, V., & Raymond, H. (2011). Gold and financial assets: are there any safe havens in bear markets. *Economics Bulletin*, *31*(2), 1613-1622.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

CFA Institute. (2017). MiFID II: A new paradigm for investment research - investor perspectives on research costs and procurement. Report. Available at: https://www.cfainstitute.org/-/media/documents/support/advocacy/mifid_ii_new-paradigm-for-research-report.ashx. [Accessed: 1 May 2023].

Das, S. P., & Padhy, S. (2018). A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting. *International Journal of Machine Learning and Cybernetics*, *9*(1), 97-111.

Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, *91*, 106263.

Dehnad, K. (2011). Behavioral finance and technical analysis. *The Capco Institute Journal of Financial Transformation*, *32*, 107-111.

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., ... & Lautenbach, S. (2013). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, *36*(1), 27-46.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, *25*(2), 383-417.

Feldstein, M. (1980). Inflation, tax rules and the stock market. *Journal of Monetary Economics*, *6*(3), 309-331.

Fix, E., & Hodges Jr, J. L. (1951). *Discriminatory analysis: nonparametric discrimination, consistency properties*. California Univ Berkeley.

Flood, M. D. (1994). Market structure and inefficiency in the foreign exchange market. *Journal of International Money and Finance*, *13*(2), 131-158.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

Goldstein, R. (1993). Conditioning diagnostics: Collinearity and weak data in regression.

Guyon, I., Bennett, K., Cawley, G., Escalante, H. J., Escalera, S., Ho, T. K., ... & Viegas, E. (2015, July). Design of the 2015 chalearn automl challenge. In *2015 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67.

Hood, M., & Malik, F. (2013). Is gold the best hedge and a safe haven under changing stock market volatility?. *Review of Financial Economics*, *22*(2), 47-52.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression. John Wiley & Sons.

Huang, C. F., & Li, H. C. (2017). An evolutionary method for financial forecasting in microscopic high-speed trading environment. *Computational Intelligence and Neuroscience*, *2017*.

Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & operations research*, *32*(10), 2513-2522.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, *374*(2065).

Kelleher, J. D., Mac Namee, B., & D'arcy, A. (2020). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press.

Kisela, P., Virdzek, T., & Vajda, V. (2015). Trading the equity curves. *Procedia Economics and Finance*, *32*, 50-55.

Kosc, K., Sakowski, P., & Ślepaczuk, R. (2019). Momentum and contrarian effects on the cryptocurrency market. *Physica A: Statistical Mechanics and its Applications*, *523*, 691-701.

Kowalski, C. J. (1972). On the effects of non-normality on the distribution of the sample product-moment correlation coefficient. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *21*(1), 1-12.

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 116659.

Lee, M. C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, *36*(8), 10896-10904.

Li, Z., Tam, V., & Yeung, L. (2016, July). Combining cloud computing, machine learning and heuristic optimization for investment opportunities forecasting. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3469-3476). IEEE.

Magdon-Ismail, M., Atiya, A. F., Pratap, A., & Abu-Mostafa, Y. S. (2004). On the maximum drawdown of a Brownian motion. *Journal of applied probability*, *41*(1), 147-161.

McCown, J. R., & Shaw, R. (2017). Investment potential and risk hedging characteristics of platinum group metals. The Quarterly Review of Economics and Finance, 63, 328–337.

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)*, *42*(2), 109-127.

Menkhoff, L. (2010). The use of technical analysis by fund managers: International evidence. *Journal of Banking & Finance*, *34*(11), 2573-2586.

Mosley, L. (2013). A balanced approach to the multi-class imbalance problem.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.

Pareek, M. K., & Thakkar, P. (2015, November). Surveying stock market portfolio optimization techniques. In *2015 5th Nirma University International Conference on Engineering (NUiCONE)* (pp. 1-5). IEEE.

Park, C. H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis?. *Journal of Economic surveys*, *21*(4), 786-826.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, *42*(1), 259-268.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

Perols, J. (2011). Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, *30*(2), 19-50.

Piasecki, K., & Stasiak, M. D. (2020). Verification of the Precious Metals Market Effectiveness–Gold and Silver.

Popat, R. R., & Chaudhary, J. (2018, May). A survey on credit card fraud detection using machine learning. In *2018 2nd international conference on trends in electronics and informatics (ICOEI)* (pp. 1120-1125). IEEE.

Pring, M. J. (2002). *Technical analysis explained: The successful investor's guide to spotting investment trends and turning points*. McGraw-Hill Professional.

Qi, M., & Wu, Y. (2006). Technical trading-rule profitability, data snooping, and reality check: Evidence from the foreign exchange market. *Journal of Money, Credit and Banking*, 2135-2158.

Radetzki, M., & Wårell, L. (2020). *A handbook of primary commodities in the global economy*. Cambridge University Press.

Rundo, F., Trenta, F., di Stallo, A. L., & Battiato, S. (2019). Machine learning for quantitative finance applications: A survey. *Applied Sciences*, *9*(24), 5574.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, *27*(3), 379-423.

Singh, R., & Srivastava, S. (2017). Stock prediction using deep learning. *Multimedia Tools and Applications*, *76*, 18569-18584.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, *25*.

Sullivan, R., Timmermann, A., & White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap. *The journal of Finance*, *54*(5), 1647-1691.

Suzuki, N., Olson, D. H., & Reilly, E. C. (2008). Developing landscape habitat models for rare amphibians with small geographic ranges: a case study of Siskiyou Mountains salamanders in the western USA. *Biodiversity and Conservation*, *17*, 2197-2218.

Suzuki, N., Olson, D. H., & Reilly, E. C. (2008). Developing landscape habitat models for rare amphibians with small geographic ranges: a case study of Siskiyou Mountains salamanders in the western USA. *Biodiversity and Conservation*, *17*, 2197-2218.

Ślepaczuk, R., & Zenkova, M. (2018). Robustness of support vector machines in algorithmic trading on cryptocurrency market. *Central European Economic Journal*, *5*(52), 186-205.

Technical Analysis Library (2018) [online] Available at: https://technical-analysis-library-in-python.readthedocs.io/en/latest/ [Accessed: 1 May 2023].

Teixeira, L. A., & De Oliveira, A. L. I. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert systems with applications*, *37*(10), 6885-6890.

Tully, E., & Lucey, B. M. (2007). A power GARCH examination of the gold market. *Research in International Business and Finance*, *21*(2), 316-325.

Worthington, A. C., & Pahlavani, M. (2007). Gold investment as an inflationary hedge: Cointegration evidence with allowance for endogenous structural breaks. *Applied Financial Economics Letters*, *3*(4), 259-262.

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26-40.

Xia, F., Zhang, W., Li, F., & Yang, Y. (2008). Ranking with decision tree. *Knowledge and information systems*, *17*, 381-395.

Yeh, C. Y., Huang, C. W., & Lee, S. J. (2011). A multiple-kernel support vector regression approach for stock market price forecasting. *Expert Systems with Applications*, *38*(3), 2177-2186.

Zarrabi, N., Snaith, S., & Coakley, J. (2017). FX technical trading rules can be profitable sometimes!. *International Review of Financial Analysis*, *49*, 113-127.

Zhang, X., Hu, Y., Xie, K., Wang, S., Ngai, E. W. T., & Liu, M. (2014). A causal feature selection algorithm for stock prediction modeling. *Neurocomputing*, *142*, 48-59.

Zhong, X., & Enke, D. (2017). A comprehensive cluster and classification mining procedure for daily stock market return forecasting. *Neurocomputing*, *267*, 152-168.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301-320.

List of Figures

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl