Ensemble Voting combined with Linearity Detection to Detect Sideways Markets in Forex

Pedro Manuel Correia Ventura
Instituto Superior Técnico, Lisboa, Portugal

***Abstract*—This work proposes an extra functionality for algorithms focused on Forex: a Sideways Markets detector. Four machine learning models were trained - Logistic Regression, Random Forest Classifier, Support Vector Classifier and XGBoost -, carrying out a vote which predicts whether the market is sideways. To train these techniques was developed an algorithm that divides the data into sections of sideways and trending markets. A third detector was designed to add robustness to these 2: a linearity detector. Furthermore, a sideways markets trading framework was conceived. Besides the difficulty of evaluating the linearity detector and the sideways classifier, since their subjective meaning and lack of evaluation metrics, the results matched the expectations presenting a high number of mutable features. Similarly, the sideways predictor showed good results, outputting a final precision of 76.73% when predicting sideways markets. To evaluate the trading framework a 7-year data set was considered, outputting a Run-Up of 13.11%, a Draw-Down of -0.21% and an ROI of 26.98%, just trading during sideways markets. Finally, combining the prediction with the trading, the overall efficacy of the work was summed on a Run-Up of 3.63%, a Draw-Down of -2.05% and an ROI of 6.75%, in the same conditions as above.**

Keywords: Sideways Markets Detector, Linearity Detector, Machine Learning, Ensemble Prediction, Forex

I. Introduction

The Forex Market is a global market for trading currencies. It is famous worldwide for being the most liquid financial market in the world and growing [1]. Trading involves time spending and is prone to human errors, emotional decisions, greed, impatience, inattention, and others. Due to these challenges, researchers started to develop automated systems, applying Machine Learning techniques to Forex Markets.

Although Forex is a very volatile and unstable market looking at a closed picture, it is possible to identify 3 types of markets on a bigger picture: uptrend, downtrend and sideways markets [2]. The first two represent trending markets, on overall increasing or decreasing movements. The third is characterized by a horizontal movement, where the price fluctuates in a tight stable range, over some time, without forming a new trend.

A. Motivation

Although uptrends and downtrends are highly studied by researchers all over the world, the sideways markets are lowly studied, representing a work opportunity. Additionally, they express new trading opportunities and may lead to potential higher profits.

B. Objectives

The main goal of this work was to develop a Sideways Detector using diverse Machine Learning predictors. This field of study was poorly developed due to its difficulty in identification, its subjective meaning and its lower profitable potential. As Sideways Markets are approximately horizontal linear graphics, a linearity detector was developed, increasing the robustness of the algorithm.

To achieve the proposed objectives of this work, some processes were implemented. Firstly, different types of data were tested to understand which was advantageous for this type of detection. Then, some data refinements were implemented. Next, the data was classified as sideways or non-sideways and used to train the machine learning techniques. Following, 4 different Machine Learning approaches - Logistic Regression, Random Forest, Support Vector Machine and XGBoost - voted to decide if the market works as a Sideways Market. Last but not least, a currency transacting framework was implemented to trade during sideways markets.

C. Main Contributions

The main achievements of this work were:

- *Linearity Detector:* Conception of 3 different linearity detectors given a slope and/or error threshold;
- *Sideways Classifier:* Implementation of a highly feature diversified algorithm to classify sideways markets on training data - used to train the machine learning predictors;
- *Automated Sideways Detector:* Combine 4 machine learning techniques to identify sideways markets;
- *Trade during Sideways markets:* Development of 2 independent trading strategies to trade during sideways markets - the first based on recognising global maximums and minimums via computing support and resistance levels, and the second based on identifying local maximums and minimums through technical indicators.

D. *Document Structure*

In Section A, literature is reviewed. In Section III, the proposed solution is described and each component is thoroughly detailed. Section IV goes over the results, analysing 3 case studies, designed to understand the strengths and weaknesses of the work. Section V concludes.

II. Background and State-of-the-Art

A **Sideways Market** is characterized by a horizontal movement, where the price fluctuates in a tight stable range over time, without forming a new trend [2].

When studying sideways markets, there are 2 central concepts to understand. The **Support Level** refers to a price level at which buying pressure is expected to be strong enough to prevent the price from falling further. It acts as a "floor" for the price, creating a barrier that prevents it from declining beyond that point. Traders often observe support levels as potential buying opportunities, as they anticipate the price to bounce back and move higher. On the other hand, **Resistance Level** represents a price level at which selling pressure is anticipated to be strong enough to prevent the price from rising further. It acts as a "ceiling" for the price, limiting its upward movement. Traders tend to view resistance levels as potential selling opportunities, expecting the price to reverse and move lower from that point.

A. *State of the Art*

Through the years, researchers all over the world spent their efforts and time implementing different techniques to find consensual results within the scientific community. Some of them found revolutionary techniques, others found small improvements on other algorithms and others just helped showing that some techniques are not useful when applied to financial markets. In this section, some of these works are explored and analysed.

First, [3] simply applied a Linear Regression method to hourly and daily forecasts, finding that the accuracy percentage of the daily data forecast is higher than the hourly data forecast. Open, High Low, Close and Volume features were used and normalized to speed up the processing time and get valuable accuracy. After that, [4] tried to implement more complex machine learning techniques to classify trend markets in Forex. They used multiple time windows to provide a higher number of features, using machine learning techniques to select and extract features. For each feature subset, 3 supervised classifiers were used: Radial Basis Function, Multilayer Perceptron and Support Vector Machine (SVM). To compare classifiers, two concepts were used: Percentage Classification Performance and Percentage Normalized Profit. They conclude that usually, the best performances occur when SVMs are included in the algorithm. Trying to get better results manipulating machine learning techniques, [5] used ensemble voting, between 4 machine learning techniques, to study the effect of resampling on the average accuracy of cryptocurrency investments. After applying some resampling methods, they concluded that resampling increased significantly the profits. On the other hand, [6] instead of just applying Machine Learning Techniques, focused on comparing basic regression models with complex deep learning models. After that, she compared the prediction using only market-rates features (Close, Open, High and Low prices) and predictions after adding newspaper headlines information. Basic models - Auto-Regressive Integrated Moving Average (ARIMA), Seasonal-ARIMA, Logistic Regression, Stochastic Gradient Descent Regressor, XGBoost Regressor and Support Vector Regressor - were compared with complex models - Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) and Gated Recurrent Unit -, concluding that the basic ones are better when fitting Forex Markets. After that, they added newspaper headline informations to the previous models, using various word embedding algorithms - Word2Vec, Bidirectional Encoder Representations from Transformers (BERT) and Financial-BERT -, concluding that the new information did not improve the performance.

After, [7] compared 2 Artificial Neural Network (ANN) structures: Elman Simple RNN with loopbacks in the hidden layer, and Multilayer Feed-Forward ANN. Structures with a variable number of hidden layers, a different number of neurons composing the hidden layer, structures with and without bias neurons and various activation functions were explored. They defined a heuristic that examines 7200 network configurations chosen at random with equal probability, and conclude about the best structure for the ANN. Also betting on ANN, [8] applied Reinforcement Learning to Forex trading, adding the concept of rewards to the algorithm. Combining a Q-system with Neural Networks, this model tried to estimate the action-value function associated with the optimal policy, maximising the expected rewards for any state. They applied early-stop, removed the outliers and subtracted bias from the features map (min, max, mean and standard deviation) to make Neural Networks useful. After several experiments, they concluded about the number of layers and that L1, L2 and Dropout regularization were not useful. The main weaknesses of this algorithm were the enormous dependence on large data sets and the lack of interpretation of a black-box Q-network. Still thinking on Neural Networks, [9] focused their work on understanding and interpreting up and down trends, through monitoring 2 experiments. The first consisted of the application of the blended learning paradigm on the networks: Vanilla-LSTM, Stacked-LSTM, Bidirectional-LSTM, CNN-LSTM and Conv-LSTM. Then, a deep network-based system used to obtain the trends of the predicted closing price of the currency pairs was proposed based on the best fit predictive networks measured using a few performance measures and Friedman non-parametric tests. Finally, the results of the two experiments were compared using technical indicators for both, short and long-term analysis, achieving an impressive accuracy between 90-100%.

Trying to add some robustness to a black-box algorithm as ANN, [10] implemented an ANN, optimizing the weights using Genetic Algorithms (GA). After several experiments, conclusions about the structure, the learning rate and the momentum of the ANN were made, choosing a 3-3-1 approach. This algorithm was implemented in order to get the lowest Root Mean Square Error, with the GA helping overtake the problem of local minimums. Was found that hybrid approaches help to face some of the ANN drawbacks. Using black-box techniques is very difficult to understand possible mistakes made by the algorithm. Thinking about that, but taking into account that GA provided some improvements in NN, [11] combined GA with SVM, applying it to the Forex Market. He used dynamic GA instead of traditional GA to avoid restarting the system when the environment changes. The work was focused on technical analysis, using SVM to classify the types of markets (uptrend, sideways and downtrend) and GA to optimize investment rules of each market type - essential to apply a good leverage strategy. The results were impressive achieving a Return of Investment (ROI) of 83%.

Almost all the work done in this area was focused on applying, managing and merging machine-learning techniques. Thinking out of the box, [12] used a completely different approach. They worked on an algorithm with 2 stages, after performing data augmentation in the currency pair data set, using technical indicators and statistical measures to perform a short-term strategy. The first stage consisted of applying a Deep Predictive Coding Network (DPCN) based on a Reptile Supervised Algorithm (RSA). The second stage consisted of applying the Higher Highs Higher Lows, and Lower Highs Lower Lows trend analysis tool. Was found that this RSA-DPCN model performs well in the Forex Market. Furthermore, this work proved the possibility of concluding about entry and exit points through the analysis of trends and trend reversals. This work shows potential to be extended not only for trend analysis, but also to measure or forecast the magnitude of trends concerning average, maximum, and minimum number units up and/or down. Other researchers, rather than apply a completely different model, just tried to implement a known model - [13] used the software ”Weka” to predict the behaviour of the Forex Market. She tried several successive experiments adding new variables that deal with Price in Percentage and successively concluded the best time to buy and sell, the best algorithm, the range of the market, and others. Finally, [14], designed a cascade model using Fundamental Data and Technical Indicator Data based on the BERT algorithm. The main idea was to extract hidden patterns from Fundamental Data, using BERT on a specific period, and aggregate them to the Technical Indicators as additional weights. Then apply again BERT to the remaining time set. Finally, the patterns were used in an NN for forecasting. Include a cascade strategy was an idea based on the real behaviour of traders. Was found that this technique outperforms all the others that were compared in forecasting strategies.

After analysing all these works, was found that there are few works focused on Sideways Markets. Some of them were focused on small improvements, like re-sampling the data, others focused on detecting trends, but very few were concentrated on detecting sideways markets. After finding this, was decided to pick up this topic, believing that it has the potential to be a step forward in this field of study.

III. Implementation

This work architecture, Figure 1, was divided into 4 main layers: Data Processing, Market Classification, Final Prediction and Trading Framework.

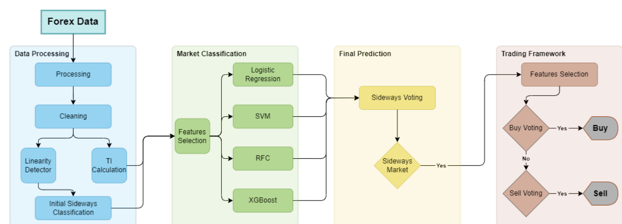


Fig. 1: Work Generic Architecture.

Firstly, a **Data Processing Layer** was applied - the extracted data was processed, applying re-sampling, cleaning and normalization, when needed. Some technical indicators were computed during this stage. Additionally, the data was divided into periods of sideways and non-sideways markets - this classification was used further to train the predictors.

Then, the **Market Classification Layer** was in charge of computing the features needed for each machine-learning predictor, as computing the methods itself. In this layer, the math behind the feature selection was computed, wherein each algorithm received as input different features selected specifically for each of them. This layer executed 4 consecutive models, applying Logistic Regression, Support Vector Machine, Random Forest and XGBoost predictors to the data.

To achieve the purpose of predicting periods of non-trending markets, the **Final Prediction Layer** received the information about the 4 applied techniques and computed an ensemble voting to detect potential sideways market entry and exit points.

Finally, the **Trading Framework Layer** applied specific trading strategies specialized on sideways markets. This layer was responsible for trading when the previous module predicted the market as sideways - it studied potential buy and sell opportunities, deciding when to open or close a trading position.

A. *Data Processing Module*

The first module developed in this work was specifically designated for processing the data and making it useful for the subsequent steps of the algorithm. Re-sampling, normalization, cleaning, and other procedures were added when necessary. Furthermore, some technical indicators were computed. Finally, one of the most important tasks of the study was completed: the segmentation of the data into sideways and non-sideways market periods - the machine learning techniques were further trained using this categorisation.

1) Data Re-Sampling: When talking about sideways markets, traders tend to use hourly sampled data [2] - every hour a new sample is released. Despite the initial trial of using hourly data, was concluded that using price-based data, "Renko10" in this case, is more effective. Unlike traditional candlestick charts, Renko charts are built using boxes that represent a fixed price movement, regardless of the time. The "10" in "Renko10" refers to the brick size, meaning that each Renko brick represents a price movement of 10 pips - a new brick appears once the price has moved 10 pips in the same direction. This re-sampling idea was based on the works of Borges [5] and Matos [15], who proved that data-based re-sampling improves the overall results for Forex.

Usually, preceding and following a period of non-trending markets, there are periods of accentuated trends, being normal to have sharp increasing or decreasing peaks near the desired entry and exit points. This change led to an improvement since the one-sample peaks that were messing up the algorithms were replaced by consecutive samples, which are easier to understand for the predictors and their typical delay.

2) Data Cleaning: The re-sampling process solved another problem that the time-based data set had - the 48 equal and non-tradable samples during the weekends. Despite that, a weekend remover feature is attached to the work code.

3) Data Normalization: Theoretically, scaling data leads to improvements in the overall performance of the machine learning classifiers [16], e.g., SVM Predictors face problems in dealing with non-scaled data. Furthermore, researchers such as Nobre et. al [17], proved that normalization outcomes in significant improvements in the results - these works were focused on predicting if the next sample represents an increase or a decrease in the price. Despite relying on different types of predictions, was decided to implement a normalization method in this work. The main idea was to transform data by scaling each feature to a given range, $\{0, 1\}$ in this case. In the general case, the data fitted the scaler and then the scaler transformed the data, i.e., the data was scaled on its own range. In other cases, the data was transformed by a scalar previously fitted by the financial market price, since there were some correlations to maintain and scaling them on their own range would destroy it.

4) Initial Sideways Classification: After downloading the data from an external provider, Dukascopy Bank [18], it was used to calculate whether a section behaves as a sideways (1) or as a non-sideways (0) market - this classification was used further to train the machine-learning methods. A sideways market is usually characterized by a horizontal price movement between two boundaries. Following this, the main idea behind the classification was to find horizontal sections and expand them. After, some control metrics were tested to approve or deny the classification. Figure 2 summarizes the idea behind this algorithm.

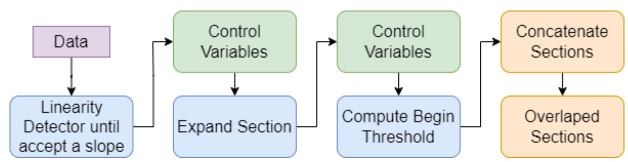


Fig. 2: Initial Classification Overall Structure.

First, applying an iteration step and a processing size, the data is iterated by a linearity detector. When it detects a slope lower than the given threshold, the section is classified as potential sideways, and are computed some variables of control. If confirmed, the section suffers expansion for both sides - the maximum and minimum currency prices of the initial section are calculated and the section is expanded for both sides, individually, until breaking one of the computed boundaries. Again, some variables of control are computed. As researchers, predictors have a normal processing delay until detecting a sideways market - the market needs to bounce for some time until considered sideways. Thinking on that, a begin threshold is computed in order to ignore the first samples of a pre-classified sideways section. Finally, the section is set as sideways (1).

To deal with consecutive nearby sections classified as sideways markets, a concatenation feature is computed - if one of the sections has his maximum and minimum limits between the maximum and minimum limits of the other, the algorithm considers them as one big section instead of two small sections. This feature just ignores sudden peaks and valleys between two nearby sideways. Furthermore, when one section is expanded inside another the algorithm re-classifies one of the overlapped sections as a non-sideways (0) section, prioritising those most horizontally stretched. Without this feature, the predictors would be deceived since they would interpret 2 independent sections as one.

5) Linearity Detector: The developed linearity detector has 3 main branches:

- *Slope Detection:* Detects the overall movement of the data, within a specific slope, ignoring the respective error [1].
- *Error Detection:* Detects straight movements on the data, within a specific error threshold, ignoring the respective inclination.
- *Complete Detection:* Detects straight specific movements on the data, within a specific slope and a specific error limit.

B. Market Classification Module

After processing, the data was ready to be used to train and test different methods. This module was in charge of selecting the feature set to train each predictor, checking the market type and executing parameter tuning.

1) Features Selection: Each machine learning predictor was trained with a specific and pre-selected set of features: price movements, technical indicators and diverse flags like maximum, minimum and mean values of specific sections. The feature space was chosen considering human reasoning and adjusted with some variables that would help the weaknesses of the existing predictions. Successively the predictors were trained and the results were analysed - each iteration of this examination concluded on the defects of the prediction, adding and removing new features to the global feature space.

The first feature space was composed of the financial data and some of the most famous technical indicators such as Bollinger Bands, RSI and 2 Moving Averages - SMA100 and SMA200. Sequentially, Lower Value Averages - SMA30 and SMA50 - were introduced to detect earlier entry and exit points, and locate exit points near sudden peaks and valleys. Additionally, Higher Value Moving Averages - SMA300 - were oriented to stretch the classifications and eliminate too-early exit points, while avoiding miss understandings near small peaks and valleys. Finally, the maximums, minimums and means were added in the sense of approaching the algorithm to the logic of expansion executed in the initial classification. After creating a robust and reliable feature space, it was important to understand which features help each individual predictor and what features just mess up them. After testing different feature sets composed of different variants of the feature space, individual feature sets were attached to each predictor - the ponderation was done between the precision of the "1" and "0" predictions and the number of predicted "1".

[1]This detector was the only one used in this work.

2) Machine Learning Predictors: After defining the feature set to train each predictor, was time to perform the predictions itself. For that, 4 machine learning classifiers were executed - Logistic Regression, Support Vector Machine, Random Forest and XGBoost - dividing the data into sections of sideways and non-sideways markets. These predictions were used further in the Final Prediction Module which ponders them and concludes about the market perspectives.

3) Parameters Tuning: Another strategy used to refine the fitted models was parameter tuning. Each Python tool has a set of specific and mutable parameters to fit the model the best possible - these parameters allow adjusting the model to the used data and the desired goal. Some of their capabilities are: specify the type of applied penalty, e.g., none, l1 penalty or l2 penalty; the maximum number of iterations taken for the solvers to converge; the maximum depth of a tree; if it is supposed to apply early stopping to find the optimal number of boosting rounds; and others. Based on the work done by Fernandes et. al [19], *Hyper Optimization* [20] was the used parameter tuning strategy - it consists of a Python library that combines a good running time with optimal result detection. Despite the prediction quality added, the real increase was small, as expected [21].

C. Final Prediction Module

This module was in charge of executing an Ensemble Voting between the 4 predictions computed previously. When classifying the data used further to train the predictors was impossible to look for single entry and exit points due to its normal and prohibitive unbalanced data, i.e., the number of classified entry and exit points would be despicable compared with the non-entry and non-exit points. Therefore, the machine learning predictors would not be capable of predicting accurately, since they do not perform on highly unbalanced data [21]. For that reason, the training process was computed by classifying the data set in compacted sections of sideways or non-sideways markets.

In this Module, the approach used to predict the markets was changed: if so far was impossible to classify the market looking to single entry and exit points, from now on, the approach is exactly that. Despite classifying the sets in sections, the algorithms look at them as single and independent points. Thus, is normal that the predictors do not find straight and compacted sections of sideways markets without noise between them. To face this problem was decided to perform a final prediction that, instead of voting point to point, looks for entry and exit points assuming about the remaining samples. Hence, the market was assumed to move sideways from the moment that the algorithm finds an entry point until it finds an exit point, and presumed non-sideways in the remaining situations - this prediction avoids noise, creating longer and compacter sideways sections.

The ensemble voting considered a point as an entry point when the market behaviours as non-sideways (0) and the majority of the forecasters predicted the point as sideways (1), i.e., a minimum of 3 votes out of 4. Oppositely, was considered an exit point when the market behaviours as sideways (1) and the majority of the forecasters predicted the point as non-sideways (0).

D. *Trading Framework Module*

The last module of this work was in charge of finding potential buy and sell opportunities by applying 2 distinct trading strategies: one focused on trading near the support and resistance levels and the other focused on trading near local maximums and minimums. Since this work is exclusively focused on sideways markets, the trading framework just trades inside these periods. Furthermore, a position is never left open - when a short position is adopted and the algorithm finds a sideways market exit point, the trading framework closes the opened position instantly. This strategy decreases the potential profits since it ignores trading near sideways markets extremities, but makes it possible to understand the effectiveness of each sideways market by comparing the initial and the final money.

1) Stop Gain and Stop Loss: Both, stop-loss and stop-gain strategies were applied during the trading process. A **stop loss** order constitutes a predetermined price threshold strategically established by the user to curtail prospective losses by triggering an automatic closure of a position when the market moves adversarially [2]. Conversely, a **stop gain** order designates a predefined price point, whereby a position is automatically closed to safeguard accrued profits, thereby preempting potential market reversals [2]. These mechanisms are instrumental in orchestrating risk management and profit optimization on financial markets.

2) Support and Resistance Levels Approach: Non-trending markets bounce between an upper limit, the resistance level, and a bottom limit, the support level. This approach is the most famous trading strategy when dealing with sideways markets [2], and simply builds on adopting a short position near the resistance level, and opting for a long position near the support level. The only challenge when applying this strategy was to define the support and resistance levels. Knowing that the algorithm would have a delay in detecting entry points, was imposed a fixed delay - 100 samples -, and the maximum and minimum values were computed from that backward point until the actual point.

3) Local Maximum and Minimum Approach: Instead of trading near the global limits, traders may find it useful to trade every time that an inversion point occurs, i.e., near local limits. To catch these points, a technical analysis was performed resorting to 3 technical indicators [22]: Bollinger Bands (BB), Relative Strength Index (RSI) and Commodity Channel Index (CCI) [2]. Given a vote threshold, these indicators performed voting applying the following rules [22]:

- *Local Maximum:* RSI >65 or CCI >165 or Upper BB <Price.
- *Local Minimum:* RSI <35 or CCI <-165 or Lower BB >Price.

IV. RESULTS

To better interpret the developed work, were included 3 case studies: initial classification, final prediction and trading framework.

A. *Financial Data*

To study the effective results, the EUR/USD currency pair was tested, in a data set containing ”Renko10” information from 01/01/2005 to 06/06/2023, combining 123,073 samples, on a train/test ratio of 80/20. Leaving apart the ”Open”, ”High” and ”Low”, the ”Close” price was the only computed in the entire work - it represents the price registered in the last moment of the time frame. Finally, the ”Bid Price” was used in the entire work, computing the ”Ask Price” as Bid Price + Spread.

A sideways market is a subjective concept, which leads to different interpretations depending on the researcher - the initial classification method was fitted to train the predictors to detect markets with a minimum length of 200 samples, tuning the remaining parameters to fit this.

B. *Evaluation Metrics*

To measure the quality of the work and to better analyse the case studies through numbers, some evaluation metrics were calculated, giving hints about the possible uses of the proposed implementation.

Since its results were subjective, Case Study A was not evaluated by metrics but based on graphical analysis. In contrast, and since Case Study B analyses a machine learning prediction, Precision, Recall and F1-Score were analysed. On the other hand, Case Study C analyses a trading strategy studying the Return Of Investment (ROI), the Number of Positive Trades, the Draw-Down and the Run-Up.

[2]Different indicators may be used, such as Moving Averages (SMA or EMA), Moving Average Convergence Divergence, Stochastic Oscillator or others. It all depends on the trader.

The metrics are described as follows:

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}. \quad (1)$$

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \quad (2)$$

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

$$ROI = \frac{\text{Final Capital} - \text{Initial Capital}}{\text{Initial Capital}} \times 100. \quad (4)$$

Draw-Down: Measures the highest loss during a trading cycle.

Run-Up: Represents the highest profit during a trading cycle.

C. *Case Study A - Initial Classification*

The first case study was reserved to understand the outcomes of the initial classification. Since this classification is subjective, there were no metrics useful to measure its quality beyond analysing graphics - regardless of that, the initial classification fited the expectations. Furthermore, the algorithm is very rich in adaptable features, making it easy for any researcher to adapt the output with its convictions. Figure 3 shows some examples of this classification.

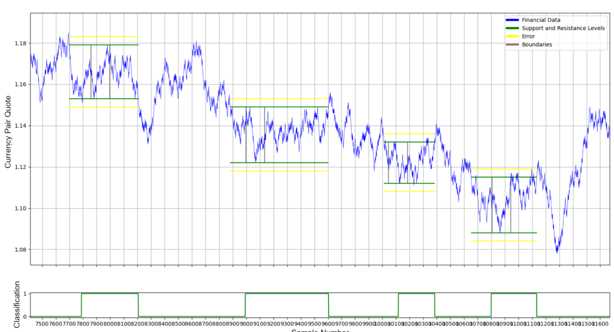


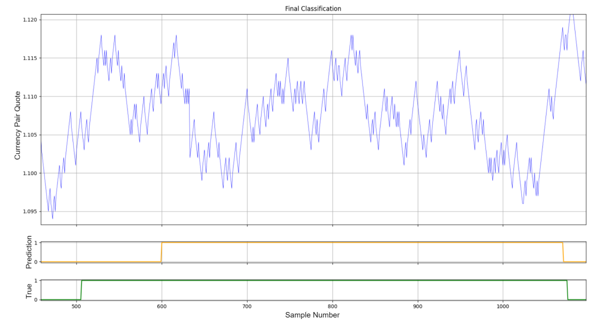
Fig. 3: Initial Classification - Sideways Example.

D. *Case Study B - Final Prediction*

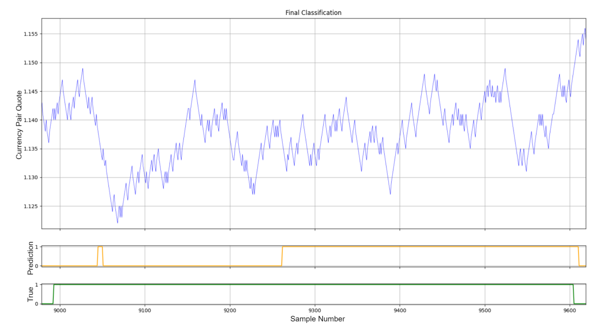
The focus of this case study was to show the results of the prediction, specifying the results after computing the 4 forecasts as one, i.e., voting. The final prediction results are displayed in Table I and some examples are displayed in Figure 4.

TABLE I: Prediction Results - Evaluation Metrics.

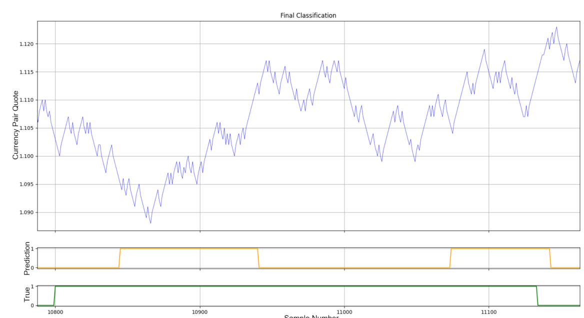
Prediction	**No. Predictions**	**No. Classifications**	**Precision**	**Recall**	**F1-Score**
0	20,305	15,714	0.72450	0.93617	0.81685
1	4,310	8,901	0.76729	0.37153	0.50064



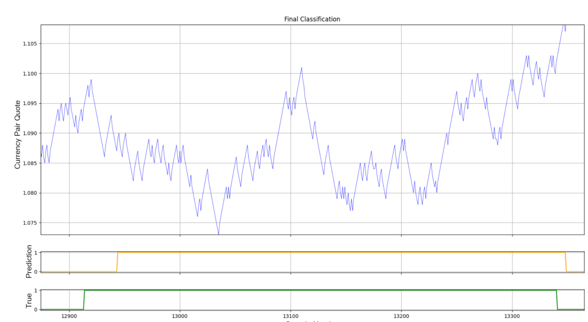
(a) Example 1.



(b) Example 2.



(c) Example 3.



(d) Example 4.

Fig. 4: Prediction Results - Graphical Representations.

To better understand the results, the strengths and weaknesses were grouped into categories separated by 3 classes: entry points, exit points and their combination, i.e., the sections themselves. Table II describes each category showing the overall performance of the predictor.

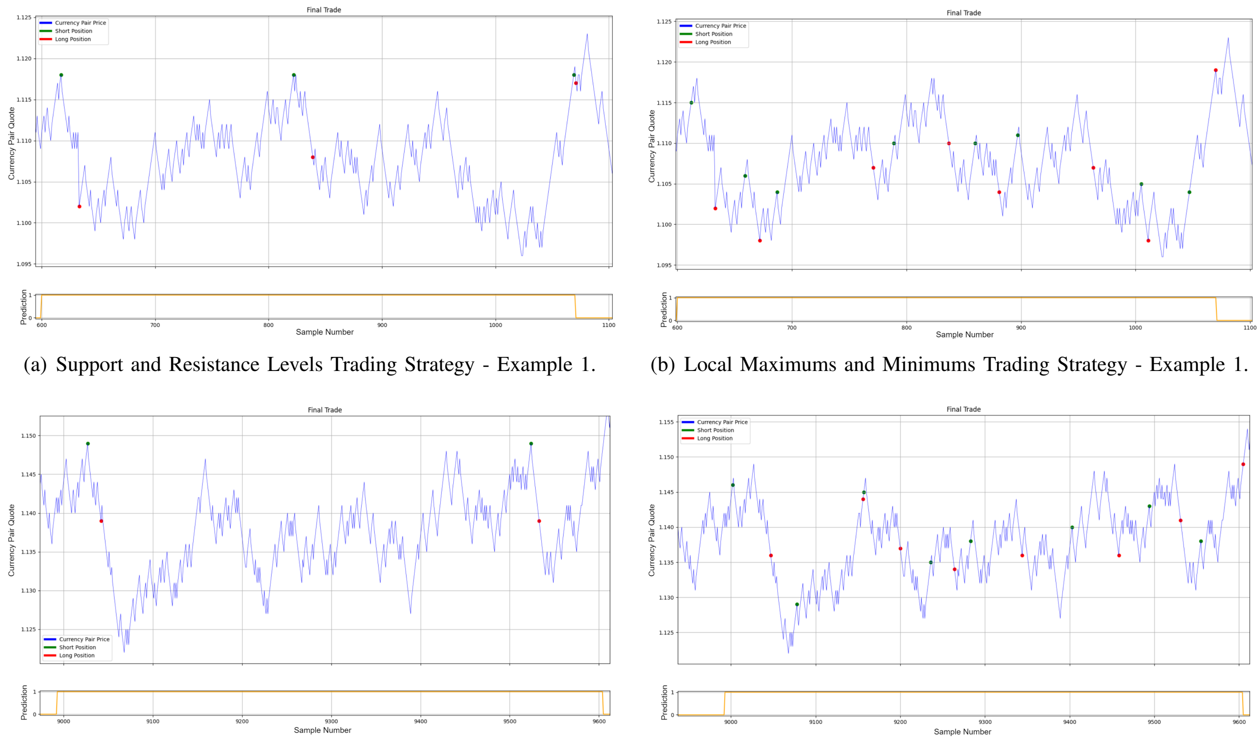
TABLE II: Prediction Results - Strengths and Weaknesses Analysis.

	Category	Definition	Occurrences [%]
Entry Point	Good	Classified and predicted points with a small difference.	29.79%
	Early	Predicted point much earlier than the classified point - represent a problem since the algorithm acts as a non-trending situation, during a trend.	4.26%
	Delayed	Classified point much earlier than the predicted point - not a problem, just decrease the potential profits.	4.26%
	Noise	Represents the entry point of a very short section - not a problem since the algorithm will not have time to open a trading position.	21.27%
	Wrong	Non-sense predictions that deceive the trader.	4.26%
	Wrong but non-trending	Finds an entry point to a non-classified but non-trending section - in practice, it consists of a small sideways market.	14.89%
	Divided	Sometimes the prediction divides a single sideways section into 2 or 3. This category represents the entry points of the second or third subsection. These points are well predicted - in reality, the classification suggests a sideways section, so, if the predictor finds a too-early exit point, it is good if then it can predict the market as a sideways market again.	21.27%
Exit Point	Good	Classified and predicted points with a small difference.	38.30%
	Early	Predicted point much earlier than the classified point - not a problem, decreases the potential profits.	31.91%
	Delayed	Classified point much earlier than the prediction point - represents a problem since the algorithm continues acting as a non-trending situation for some time.	4.26%
	Noise	Represents the exit point of a concise section - not a problem since the algorithm will not have time to open a trading position.	21.27%
	Wrong	Exit points after a ”Wrong” entry point. It cannot be classified as a good or bad prediction since the error was in the entry point prediction and inevitably the algorithm will have to find an exit point. This set was called ”Wrong” since the algorithm should have found an earlier exit point and the wrong entry point would have been classified as ”Noise” and non-dangerous instead of ”Wrong” and dangerous section.	4.26%
Sections	Good	With good entry and exit points.	17.02%
	Wrong	Random and unsubstantiated prediction - represent a problem since they are deceiving the trader.	4.26%
	Non-Classified	Non-classified but non-trending section - in practice, it consists of a small sideways market.	17.02%
	Non-Detected	The prediction did not find the classified section - not a problem, just decreases the potential profit.	9 occurrences
	Noise	Classifications with a very low length - not a problem since a short section is non-tradable, i.e., the algorithm will not have time to open a trading position.	21.27%
	Divided	Represents the single classified sections divided into multiple predicted sections - imperfect but correct predictions.	40.43%

Analysing **Entry Points** was concluded that ”Good”, ”Delayed”, ”Wrong but non-trending” and ”Divided” points, representing correct tradable points, encompass 70.21% of the predictions. ”Early” and ”Wrong” points represent incorrect trading entry points and encompass 8.52% of the predictions. Finally, 21.27% of the points were considered as ”Noise” being ignored by the trading framework. Analysing **Exit Points**, ”Good” and ”Early” points represent correct tradable points and encompass 70.21% of the predictions. ”Wrong” and ”Delayed” points represent incorrect trading points and encompass 8.52% of the predictions. Finally, 21.27% of the points were considered as ”Noise” being ignored by the trading framework. Analysing the **Overall Sections**, ”Good”, ”Non-classified” and ”Divided” sections represent correct tradable sections, and encompass 74.47% of the predictions. ”Wrong” sections represent incorrect trading sections and encompass 4.26% of the predictions. Additionally, 21.27% of the sections were considered as ”Noise” being ignored by the trading framework. Finally, 9 classified sections were not detected by the predictors.

E. Case Study C - Trading Framework

The last case study aims to analyse the trading framework, measuring its capacity during sideways markets. The initial money was 100,000 EUR, and the considered broker was the Dukascopy Bank [18]. It was considered a spread of 0.0001, a stop loss of 0.010 and a stop gain of 0.015. This bank charges 20US$ for each 1,000,000US$ spent [23]. Figure 5 shows one example of the trading process on predicted sideways markets. Comparing them, it is possible to understand that the first looks exclusively for support and resistance levels, trading fewer times, and the second tries to detect all the local maximums and minimums, trading more often.



(a) Support and Resistance Levels Trading Strategy - Example 1.

(b) Local Maximums and Minimums Trading Strategy - Example 1.

(c) Support and Resistance Levels Trading Strategy - Example 2.

(d) Local Maximums and Minimums Trading Strategy - Example 2.

Fig. 5: Comparison between trading, in the same sideways section, using different approaches.

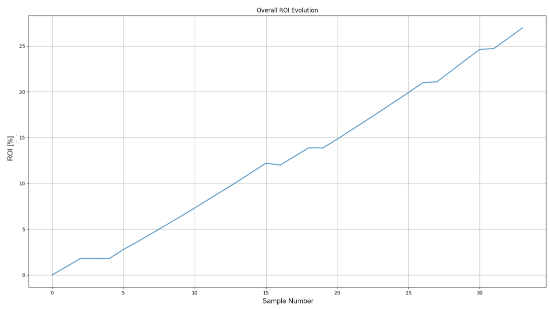
TABLE III: Trading Results - Trading Framework and Overall Work Performances.

Study Type	Trading Points	ROI	ROI per Trade	Positive Trades	Draw-Down	Run-Up	No. Trades
Trading Framework	**Support and Resistance Levels**	26.98%	0.82%	87.88%	-0.21%	13.11%	33
	Local Max and Min (1 vote)	19.25%	0.14%	63.70%	-2.24%	3.40%	135
	Local Max and Min (2 votes)	21.87%	0.18%	63.64%	-2.46%	4.63%	121
	Local Max and Min (3 votes)	17.23%	0.30%	67.24%	-1.94%	4.35%	58
Overall Work	**Support and Resistance Levels**	6.75%	0.18%	48.65%	-2.05%	3.63%	37
	Local Max and Min (1 vote)	4.46%	0.06%	57.53%	-4.45%	4.48%	73
	Local Max and Min (2 votes)	0.04%	0.00%	52.46%	-4.20%	4.81%	61
	Local Max and Min (3 votes)	-2.72%	-0.09%	43.33%	-3.39%	2.75%	30

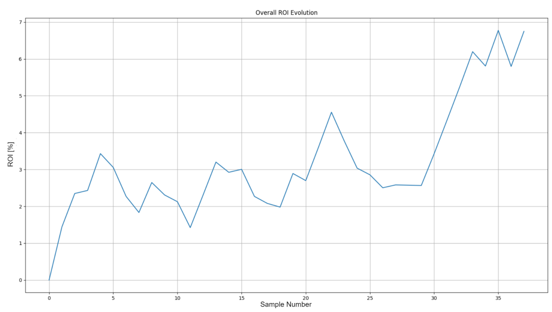
1) Trading Framework Performance: Firstly, the effectiveness of the trading framework was studied, trading on the ”perfect” initially classified sections. Figure 6(a) shows its effectiveness, applying the most effective strategy, with a straightly increasing ROI with small and occasional decreases. Table III displays its results.

As anticipated the best performance occurs when applying the Support and Resistance Levels Approach since it shows better values of ROI, ROI per Trade, % of positive trades, Run-Up and Draw-Down. The best-obtained ROI and ROI per Trade were, respectively, 26.98% and 0.82%. When looking at the local maximums and minimums studies, some conclusions were taken: despite the ROI being similar on the 3, the ROI per Trade on the last study was better than the others.

2) Overall Work Performance: Finally, the effectiveness of the entire work was studied, trading on the predicted sections. Figure 6(b) shows an irregular, but positive ROI evolution, suffering from the accumulated error on the market prediction. Table III summarizes the quality of the work. As anticipated the best performance occurs applying the Support and Resistance Levels Approach since it shows higher values of ROI, ROI per Trade and Draw-Down, despite it showing just the third best positive/negative trade rate and no. of trades, and the second worst Run-Up. The best ROI and ROI per Trade were, respectively, 6.75% and 0.18%. Looking at the local maximums and minimums studies, the 1 vote strategy was the better: it displayed a better ROI, ROI per Trade and no. of trades.



(a) Trading framework performance - classified data.



(b) Overall work performance - predicted data.

Fig. 6: ROI Evolution - Support and Resistance Strategy.

V. Conclusions

During this work, 4 features were developed: a linearity detector, a sideways market classifier, a sideways markets forecaster and a trading framework. The final ROI was 6.75%, resulting from a prediction precision of 76.73% combined with a trading ROI of 26.98%.

A. Future Work

Due to its innovative approach, and despite significant and relevant developments, there are plenty of things to do in the future:

- *Parameters Tuning:* Adjust the initial classification to detect different markets;
- *Approximate the Classification Methods:* Classify using the features used to predict instead of classifying through linearity detection;
- *Machine Learning Methods:* Future researchers may find it useful to use different machine learning or even some deep learning techniques;
- *Weight the Prediction:* Specialize each predictor in detecting entry or exit points, instead of both;
- *Incorporate the Linearity Detector on the Final Prediction:* While the predictors are looking for entry or exit points, the Linearity Detector would complement with information about the linearity;
- *Trading Framework Improvements:* Merge the implemented strategies or use different ones;
- *Trading Framework Around Extremities:* Trade on the edges of sideways markets, instead of trading exclusively during these markets.

References

[1] G. Cespa, A. Gargano, S. J. Riddiough, and L. Sarno, “Foreign exchange volume,” *The Review of Financial Studies*, vol. Vol. 35, May 2022.

[2] G. Soros, *The Alchemy of Finance*. John Wiley & Sons Inc., 1986.

[3] M. Sarkar and U. Ali, “Eur/usd exchange rate prediction using machine learning,” *I. J. Mathematical Sciences and Computing*, 2022.

[4] A. A. Baasher and M. W. Fakr, “Forex trend classification using machine learning techniques,” Master’s thesis, Arab Academy for Science and Technology, 2011.

[5] T. A. Borges and R. F. Neves, “Ensemble of machine learning algorithms for cryptocurrency investment with different data resampling methods,” Master’s thesis, IST, 2020.

[6] S. Atha and B. Bolla, “Do deep learning models and news headlines outperform conventional prediction techniques on forex?,” Master’s thesis, Liverpool John Moores University, 2022.

[7] M. A. H. Ismail, Z. Husin, M. L. Yasruddin, and W. K. Tan, “Automated trading system for forecasting the foreign exchange market using technical analysis indicators and artificial neural network,” 2022.

[8] J. Carapuço, R. Neves, and N. Horta, “Reinforcement learning applied to forex trading,” *Applied Soft Computing Journal*, 2018.

[9] A. K. Das, D. Mishra, K. Das, A. K. Mohanty, M. A. Mohammed, A. S. Al-Waisy, S. Kadry, and J. Kim, “A deep network-based trade and trend analysis system to observe entry and exit points in the forex market,” *Mathematics*, 2022.

[10] P. K. Sarangi, M. Chawla, P. Ghosh, S. Singh, and P. Singh, “Forex trend analysis using machine learning techniques: Inr vs usd currency exchange rate using ann-ga hybrid approach,” *Materials Today: Proceedings*, 2022.

[11] B. J. de Almeida, “Combining support vector machine with genetic algorithms to optimize investments in forex markets with high leverage,” Master’s thesis, IST, 2016.

[12] S. Dash, P. Sahu, D. Mishra, P. Mallick, B. Sharma, M. Zymbler, and S. Kumar, “A novel algorithmic forex trade and trend analysis framework based on deep predictive coding network optimized with reptile search algorithm,” *Axioms*, 2022.

[13] L. Abednego and C. E. Nugraheni, “Forex data analysing using weka,” 2020.

[14] A. Pornwattanavichai, S. Maneeroj, and S. Boonsiri, “Bertforex: Cascading model for forex market forecasting using fundamental and technical indicator data based on bert,” *Digital Object Identifier*, 2022.

[15] R. F. N. Diogo Mourato de Matos, “Ensemble of machine learning methods with genetic algorithms and re-sampling for forex market trading,” Master’s thesis, IST, 2022.

[16] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, and Z. Siddique, “Effect of data scaling methods on machine learning algorithms and model performance,” *Technologies*, 2021.

[17] J. Nobre and R. F. Neves, “Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets,” Master’s thesis, IST, 2018.

[18] D. Bank. https://www.dukascopy.com, 2023. Last accessed 12 Apr 2023.

[19] P. Fernandes, “Cryptocurrency price direction prediction through ensembles of machine learning algorithms allied with percentage resampling,” Master’s thesis, IST, 2022.

[20] J. Bergstra, D. Yamins, and D. Cox, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures.” TProc. of the 30th International Conference on Machine Learning, 2013.

[21] M. A. Junior, P. Appiahene, O. Appiah, and C. N. Bombie, “Forex market forecasting using machine learning: Systematic literature review and meta-analysis,” *Journal of Big Data*, 2023.

[22] R. J. B. Jr. and J. R. Dahliquist, *Technical Market Indicators: Analysis & Performance*. John Wiley & Sons Inc., 1998.

[23] M. popular instruments on SWFX Swiss FX & CFD Marketplace, “Dukascopy bank..” https://shorturl.at/aDX28, 2023. Last accessed 26 Aug 2023.