

# Market Prices Trend Forecasting Supported By Elliott Wave's Theory

Tomas Vantuch<sup>1</sup>, Ivan Zelinka<sup>1</sup> and Pandian Vasant<sup>2</sup>

<sup>1</sup> Department of Computer Science, VSB-Technical University of Ostrava, 17. listopadu 15 708 33, Ostrava-Poruba, Czech Republic.

`tomas.vantuch@vsb.cz`

`ivan.zelinka@vsb.cz`

<sup>2</sup> Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Malaysia

**Abstract.** The forecasting of the stock markets' trends is one of the most frequently applied point of interests in machine learning (ML) industry from its beginning. The theory of Elliott waves' (EW) patterns based on Fibonacci's ratios is also heavily applied in several trading strategies and tools which are available on the market and also there are many studies based on analysis and application of those patterns. This paper covers market's trend prediction by ML algorithms such as Random Forest and Support Vector Machine. The trend prediction is supported by application of recognized Elliot waves which was performed by custom developed algorithm based on available knowledge about the patterns. The combination of ML algorithms and EW pattern detector achieved significantly higher performance compare to the ML algorithms only.

**Key words:** Elliott waves, Fibonacci ratios, Support Vector Machine, Random Forest, Stock Markets

## 1 Introduction

The behavior of stock market price is a reflection of complex interactions of multiple underlined processes like political statements, macroeconomic situation, news, investors psychology, previous behavior of the price itself, etc. This interactions bring huge amount of noise, uncertainty and non-rational moves into the price progress which makes it difficult to forecast.

There are many high quality studies aiming into this topic with application of ML. The movement prediction with widely known Support Vector Machine obtained interesting results in study of Huang et. al. [?]. The application of neuro-fuzzy system in the study of Atsalakis et. al [?] brought results of comparable quality. The next proposal combining the Genetic Algorithm and Support Vector Machine was tested also with success in study of Choudhry [?]. All of these models faced with different conditions (time series length, chosen markets,

length of forecast, etc.) but most of them comes with similar results which con-firms the difficulty of forecasting of the stock market direction. The application of Elliott waves, recognized by Artificial Neural network proposed by Volna et. al. [?] includes several tests for its predictability performance as well.

The extensive review of studies based on neural networks and fuzzy systems [?] concludes about significant outperforming of soft computing techniques over statistical approaches in stock price predictions.

This paper is focused on the application of Elliott wave theory on stock market data. The motivation is to develop and verify an algorithm for pattern recognition based on EW's rules (section 1.1). The extracted EW patterns will serve as input values for the trend prediction models based on machine learning algorithms. In case of success, this model can be applied as supportive tool in automatic trading software.

## 2 Experiment design

The first part of the experiment focuses on EW pattern extraction and evaluation of the few basic statistical features of the found patterns. Next part is an evaluation of the impact on the predictable performance of the ML models by application of extracted patterns. The Support Vector Machine (SVM) and Random Forest (RF) were the applied algorithms for this trend prediction. In-put of these algorithms were price values of the incomplete EW patterns 1-2-3-4 and 1-2-3-4-5. The output value of the algorithms were only binary values where 1/0 stands for prediction of up/down-trend (average price of  $n$ -future ticks is higher/lower than price of last wave). The value of  $n$  is calculated according to the EW rules, where length of forecasted fifth wave should be approximately 1.62 times longer than the first wave (in case of input of 1-2-3-4 waves) and the first corrective wave A should be of similar size as a wave 5 (in case of input of 1-2-3-4-5 waves). There is no guarantee that detected waves (first four or five) will complete the entire EW cycle, which is one of the main reason why the final performance will be decreased. The usefulness of EW pattern application will be compared to prediction based on random selection of the observation as the input value.

### 2.1 Data acquisition

The stock market data were obtained from public sources. The data is in format of one-minute measurement for MetaTrader platform [?]. It contains the price behavior of GOLD (XAG/USD) from 2011-01-01 to 2015-12-31, SILVER (XAU/USD) from 2011-01-01 to 2015-12-31 and currency pair EURUSD (EUR/ USD) from 2005-01-01 to 2015-12-31. These market titles were chosen because the higher amount of trades performed on these titles and free availability of data to study.

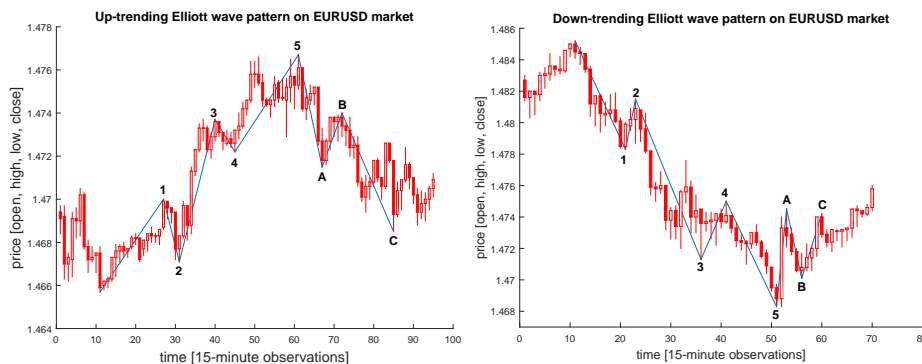
Each observation contains open, high, low, close price and traded volume of the commodity. The time series contain more than 2 millions observations in

case of GOLD and SILVER and in case of the currency pair, it was more than 4 million observations (ticks). Because of the assumption of possible application of EW's fractality (ability to be recognized on all of the time-scales), the time series were transformed also into 10-minute, 15-minute and 1-hour time series.

### 2.2 Elliott wave theory

R.N.Elliott proposed the theory in 1946 [?], which describes how every stock market repeats itself in predictable patterns, so called waves. His analysis defined the basic pattern of five motive and three corrective waves, which appears in both trend directions, bearish and bullish. These patterns posse the feature of "self-similarity", which means that the same pattern can be observed in sub-parts of the entire EW. This feature is a sign of fractal behavior of the EW's pattern. The form of the waves is very consistent and it follows Fibonacci's ratios, which is one of the major guide line of their detection.

This proposal of Elliott waves (EW) is still an active area of study and it was reviewed and extended by many authors [?, ?]. The EW's patterns were many times the motivation of application of algorithmic pattern recognition based on fuzzy logic or artificial neural networks [?, ?]. The proper identification of the actual wave can be considered as the ideal tool for prediction of the future price move and final profit target placement [?, ?].



**Fig. 1.** Up-trending (right) and down-trending (left) waves recognized on EURUSD market with 15 minute time scale and 10% tolerance

For proper indentification of the EW pattern, it has to be considered three major rules [?, ?]

1. The second wave never ends below the starting point of the first wave.
2. The third wave is never the shortest wave.
3. The wave four never ends in the price are of the fist wave.

and rules based on Fibonacci ratio's.

1. The length of wave 2 is from 50% to 62% of the wave 1.
2. The wave 3 can be 1.62, 2.62 or 4.25 longer than wave 1.
3. The wave 4 is retracing from 24% to 38% of the wave 3.
4. The wave 5 is the multiplication of 1, 1.62 or 2.62 and wave 1.
5. The wave A is almost equally length as the wave 5.
6. The wave B is retracing from 50% to 76.8% of the wave A.
7. The wave C can be length of 0.62, 1 or 1.62 times wave A.

### 2.3 Elliott wave detection

For purpose of this experiment, we developed a EW pattern detection algorithm based on EW detection rules. There are several very important variables that serves as inputs for the entire algorithm. The first of them is the vector  $X$  which is the non-stationary, trend-full time set. The time series should not be pre-processed, de-noised or filtered, because the algorithm attempts to find EV's of all of the possible sizes. The  $X$  is divided into smaller windows of length

$w$ . For each of the window, there is initiated parallel process which can run independently. The second input variable is the tolerance vector  $t$ . This vector of double values represents tolerances for the wave-selection process.

At the start of the process there are initialized two vectors  $L_{min}$  and  $L_{max}$  which hold the indexes of the local minimums and maximums of the input time set. These kind of extremes are titled as "Initiating Fractals" (IF) and firstly were introduced in Dr. Bill Williams' book "Trading Chaos" [?]. The basic definition of IFs deals with defined width (number of ticks accounted into structure) and if the high in the middle is the highest high of the structure, than it is 'up' fractal, otherwise 'down' fractal (see in Fig. ??). The process than iterates the IFs multiple times with attempt to find the best combinations which could probably represent the EWs.

**Fig. 2.** Up initiating fractals examples, source "Trading Chaos" [?]



**Pattern selection** Up-trending EW starts in the down IF, therefore the process iterates all of the indexes from  $L_{min}$  and applies them as IFs of the EW. On the other hand the down-trending EW starts in the up IF, so the process iterates  $L_{max}$  as its IFs.

The process comes through both local extremes to complete the entire wave pattern and in every time the process looks for the best matching opposite

extreme which can fit to the defined rules with adjusted level of tolerance. The application of tolerances can be generically described by following equations.

$$I_{up} = W_j \times (F_{j+1} \times (1 + t_{j+1})) \quad (1)$$

$$I_{down} = W_j \times (F_{j+1} \times (1 - t_{j+1})) \quad (2)$$

Where  $I_{up}$  and  $I_{down}$  is a couple of upper and lower boundaries for the following wave,  $W$  is the price-level of the  $j$ -th wave,  $F$  is the applied Fibonacci ratio (according to the rules) and  $t$  is the value of defined tolerance for the upcoming wave.

Every time the fractal structure can be connected to more possible fractal structures and all of this combinations are stored in stack and used in next step as valid sub-waves. When the fractal can not connect to its next follower, the entire sub-wave is deleted from the stack. During every step, some of the sub-waves which can grow multiple ways are multiplied and others, which are not able to grow anywhere are removed. When the last fractal is added into the sub-pattern, the complete EW is found (under the certain level of tolerance) and it is saved in the output matrix  $M$ . In the end of the process, the matrix  $M$  can contain similar and overlapping patterns which has to be handled before the process terminates.

**Filtering the redundancies and overlapped patterns** Rows of the matrix  $M$  represent the found EW patterns and columns are the indexes of each of the waves (including the starting point). Some of the rows differs only in few indexes and some of the rows represents patterns which are overlapped. From every group of similar or overlapped patterns it should be saved only one which fits the EW pattern's conditions the most.

It is computed the cost function of each of the pattern, which is the averaged difference in percentage between found (real) Fibonacci ratio ( $F_r$ ) and target level of Fibonacci ratio ( $F_t$ ). In cases of applied ranges instead of target levels, the difference outside of the range is taken into account only. The equation of the fitness value is as follows

$$cf(W) = \frac{1}{n} \sum_{i=1}^n \begin{cases} \min\left(\frac{|F_{t_1}-F_r|}{F_{t_1}}, \dots, \frac{|F_{t_m}-F_r|}{F_{t_m}}\right), & \text{rules 2, 4 and 7.} \\ \begin{cases} |F_l - F_r|/F_l, & \text{if } F_r < F_l \\ |F_h - F_r|/F_h, & \text{if } F_r > F_h \\ 0, & \text{if } F_l \leq F_r \leq F_h \end{cases} & \text{rules 1, 3, 5 and 6.} \end{cases} \quad (3)$$

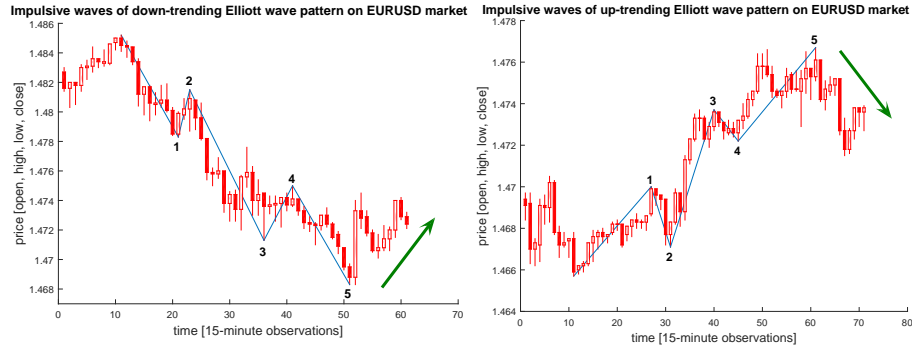
where  $F_l$  and  $F_h$  is the lower and higher border of the defined range respectively.  $F_{t_m}$  is the  $m$ -th defined target level (all of them are checked) of Fibonacci ratio. The patterns are sorted according to their cost value. The best pattern is stored and all similar and overlapping patterns are removed.

If the window contains more patterns, which are not overlapped and fits the rules under defined tolerance, algorithm returns all of them.

## 2.4 Binary classification for trend direction prediction

As it was mentioned before, the ML algorithms predict the trend from five or four input values, which represents the prices of wave ends (see in Figure ??).

The output class represents the following trend direction after the last input wave.



**Fig. 3.** Up-trending (right) and down-trending (left) waves recognized on EURUSD market with 15 minute time scale and 10% tolerance

**Support Vector Machine** The SVM was introduced by Vapnik in 1995 [?, ?] as a classifier for two-class (binary) problems and from this time it was applied in many studies for stock market prediction [?, ?] or classification [?].

In basic manner, SVM creates mapping for input vector  $x$  into the high dimensional feature space where all of the training observations are separable by designed Optimal Separating Hyperplane (OSH). OSH should maximize the distances (so called margins) between the nearest observations and the OSH itself. The observations of one class should be ideally on one of the sides of the space divided by OSH and observations of the second class should be on the opposite side of the OSH. The training observations, which are the closes to the OSH are called as support vectors.

In case of the observations are lineary separable, the solution (OSH) is given by following equation

$$Y = \text{sign}\left(\sum_{i=1}^N y_i \alpha_i (x \cdot x_i) + b\right) \quad (4)$$

where  $Y$  is the output value,  $y_i$  is the target value and  $\cdot$  represents the product of input vector  $x_i$  and  $N$  support vectors. The  $\alpha$  and  $b$  represent the parameters of the hyperplane.

In other (non-linear) cases, the SVM applies a kernel transformation function ( $K$ ) which transforms the inputs into high-dimensional feature space, which gives the ability to find suitable OSH.

$$Y = \text{sign}\left(\sum_{i=1}^N y_i \alpha_i K(x, x_i) + b\right) \quad (5)$$

There are several known kernel functions which are able to be applied for example polynomial kernel function ( $K(x, x_i) = (x \cdot x_i + 1)^d$ ), Gaussian radial bases kernel function ( $K(x, x_i) = \exp(-1/\delta^2(x - x_i)^2)$ ), etc.

Training of the SVM is equal to solve a linearly constrained quadratic problem (QP) where number of variables is equal to the number of input vectors of the training dataset.

**Random Forest** Random Decision Forest is a general title for ensemble based machine learning model which was proposed by Breiman in 2001 [?]. This model was successfully applied in many machine learning studies [?, ?]. The core idea of the algorithm is focused on application of an ensemble of CART-like tree classifiers (boosting) and their learning performed on the boosted-aggregated observations (bagging).

The decision tree (DT) is a tree-like structure of conditions with binary output values [?]. These conditions represent the nodes and leaves of the tree and serve as conditions for classification of the observation. Each condition makes a single decision on one chosen attribute from the dataset and such an attribute is called splitting criteria. The attribute becomes the splitting criteria, when its information gain value is the highest on the particular subset of observations.

Learning of the ensemble of trees means to train the set of trees where each of them obtains a different random subset of observation and different random subset of variables. This process minimizes the correlation between the trees, which increases the robustness of the model and decreases the possible amount of over-fitting. The final classification is derived from voting mechanism where votes from all of the trees are taken into account and final class is assigned to the observation by votes of the majority of the ensemble.

### 3 Results

The first part of the experiment was focused on EW pattern selection. The patterns were extracted from windows of length of 1000 observations from time series of ten-minute data and time series of hourly data. The different conditions were applied on the width of the initial fractal. The width of IF was set to 5 observations in case of hourly time series and 9 observations in case 10-minute time series.

All of the EW extractions were proceeded with four levels of tolerances (10%, 15%, 20%, 25% for all of the Fibonacci ratios) and only complete EW were taken into account (all of the waves should fit the defined criteria under the given level of tolerance). The hypothesis of this experiment was focused on how the quality and amount of found EW's depends on adjusted tolerances. The amount of found EW patterns is simply the count of patterns which fulfill the criteria and not

over-lap the other selected patterns. EW's which are inside of some longer EW pattern were not recognized in this experiment.

The median of the EW's length is also observed. The longer EW is recognized, the easier opportunity is to open the position for a trade in the matter of time. The last statistic that was computed was the Market's coverage, which simply means the ratio between observations that were members of found pattern to observations without membership to any EW. The higher amount of correctly recognized patterns implies the higher coverage, which leads to higher number of trading opportunities in the market.

| time series           |        | 10 min observations (initial fractal width = 9) |       |        |       |        |       |        |       |
|-----------------------|--------|---|-------|--------|-------|--------|-------|--------|-------|
|                       |        | 10%   |       | 15%    |       | 20%    |       | 25%    |       |
| tolerance             |        | N   | AC    | N      | AC    | N      | AC    | N      | AC    |
| Up trending pattern   | GOLD   | 14  | 5.05% | 61     | 7.39% | 120    | 8.59% | 162    | 9.06% |
|                       | SILVER | 14  | 5.01% | 55     | 6.93% | 107    | 8.23% | 166    | 9.09% |
|                       | EURUSD | 44  | 5.38% | 159    | 6.86% | 303    | 8.41% | 411    | 8.98% |
| Down trending pattern | GOLD   | 18  | 6.11% | 69     | 7.02% | 120    | 8.15% | 174    | 8.82% |
|                       | SILVER | 13  | 5.09% | 67     | 7.30% | 123    | 8.55% | 169    | 9.35% |
|                       | EURUSD | 43  | 4.68% | 151    | 6.69% | 305    | 8.02% | 398    | 8.56% |
| Average length        | GOLD   | 297   |       | 228    |       | 207    |       | 203    |       |
|                       | SILVER | 281   |       | 248    |       | 232.5  |       | 207    |       |
|                       | EURUSD | 272   |       | 203    |       | 179    |       | 165    |       |
| Market's coverage     | GOLD   | 4.19%   |       | 15.21% |       | 26.14% |       | 36.10% |       |
|                       | SILVER | 3.67%   |       | 13.78% |       | 25.33% |       | 34.81% |       |
|                       | EURUSD | 6.52  |       | 19.54  |       | 34.72  |       | 43.03% |       |

**Table 1.** Description of the quality and amount of founded EW patterns in 10-minute stock market data. The time series described GOLD, SILVER and EURUSD markets. For each of the market, there is number of found waves (N) and their average cost value (AC).

Tables (?? and ??) describe the numbers and quality of found waves and other simple statistic according to the set level of tolerance.

The number of found patters increased with higher level of tolerance but on the other hand, the higher number of waves obtained lower quality according to their cost function. The average length of found waves also differs with different applied tolerances. Higher number of found patters implied in all of the cases the higher Market's coverage.

The second part of the experiment was the extraction of incomplete EW patterns (1-2-3-4 and 1-2-3-4-5) which were later applied for attempt to predict the future trend by ML algorithms. The algorithms were adjusted experimentally according to their previous applications.

The number of observations for classification differs for each of the market. In case of the GOLD title, there were 840 observations of 4-wave pattern and 699 observation of 5-wave pattern and in case of the SILVER title, there were 772



| time series           |        | hourly observations (initial fractal width = 5) |       |        |       |        |       |        |       |
|-----------------------|--------|---|-------|--------|-------|--------|-------|--------|-------|
| tolerance             |        | 10%   |       | 15%    |       | 20%    |       | 25%    |       |
|                       |        | N   | AC    | N      | AC    | N      | AC    | N      | AC    |
| Up trending pattern   | GOLD   | 5   | 6.31% | 17     | 7.65% | 29     | 7.06% | 41     | 7.84% |
|                       | SILVER | 5   | 3.37% | 16     | 4.75% | 27     | 6.33% | 31     | 6.96% |
|                       | EURUSD | 7   | 5.59% | 25     | 7.34% | 43     | 8.12% | 55     | 8.24% |
| Down trending pattern | GOLD   | 4   | 4.97% | 10     | 5.51% | 28     | 8.77% | 39     | 9.46% |
|                       | SILVER | 5   | 7.50% | 20     | 8.27% | 31     | 8.24% | 50     | 9.67% |
|                       | EURUSD | 9   | 4.78% | 30     | 7.01% | 49     | 8.01% | 78     | 9.06% |
| Average length        | GOLD   | 644   |       | 266    |       | 237    |       | 184.5  |       |
|                       | SILVER | 260   |       | 216    |       | 199    |       | 188    |       |
|                       | EURUSD | 439.5   |       | 254    |       | 213.5  |       | 181    |       |
| Market's coverage     | GOLD   | 11.17%  |       | 21.44% |       | 35.93% |       | 45.61% |       |
|                       | SILVER | 6.19%   |       | 23.41% |       | 33.73% |       | 47.27% |       |
|                       | EURUSD | 10.41%  |       | 26.04% |       | 36.48% |       | 46.34% |       |

**Table 2.** Description of the quality and amount of founded EW patterns in 10-minute stock market data. The time series described GOLD, SILVER and EURUSD markets. For each of the market, there is number of waves (N) and their average cost value (AC).

observations of 4-wave pattern and 642 observation of 5-wave pattern. Because the EURUSD time set contained the longest observed time-range (more than 4 millions of observations represents 10-year's data) the number of incomplete patterns was higher. The number of EURUSD's 4-wave patterns was 1738 and 1464 5-wave patterns.

The cross-validation method [?] was applied for testing of classification performance and the results covers calculated values of accuracy, precision, recall, specificity and basic f-score (see in Table ??).

The results between SVM and RF algorithms obtained minimal differences and there is no motivation to compare their performance. The more interesting comparison is the increase of the performance between the application of randomly selected observations (with similar distances as EW's patterns) and application of extracted EW patterns. As we can see, when the all of the impulsive waves were detected, the trend direction was easier to predict. On the other hand, the incomplete impulsive wave pattern (1-2-3-4) had no difference in progress from the random price picks.

## 4 Conclusions

The Elliott wave theory was considered as a main idea of this paper and the algorithm for EW's pattern detection was developed and tested. The pattern matching is based on the fundamental knowledge described in previous studies which applies the counting of Fibonacci ratio sequences.

The obtained time series contained significant level of noise and uncertainty, because the stock market as a source mechanism often performs a stochastic

| trend direction prediction |             | (1-2-3-4) waves input |          |        | (1-2-3-4-5) waves input |          |        |
|----------------------------|-------------|-----------------------|----------|--------|-------------------------|----------|--------|
|                            |             | EW + RF               | EW + SVM | Random | EW + RF                 | EW + SVM | Random |
| GOLD                       | accuracy    | 52.26%                | 54.4%    | 50.29% | 64.89%                  | 68.06%   | 50.14% |
|                            | recall      | 45.05%                | 55.33%   | 35.99% | 67.66%                  | 69.05%   | 40.56% |
|                            | specificity | 59.42%                | 53.6%    | 63.17% | 62.22%                  | 67.15%   | 64.7%  |
|                            | precision   | 49.93%                | 51.41%   | 47.54% | 62.71%                  | 66.24%   | 50.44% |
|                            | f-score     | 46.86%                | 53.13%   | 40.84% | 65.06%                  | 65.51%   | 43.24% |
| SILVER                     | accuracy    | 48.18%                | 54.93%   | 50.29% | 67.5%                   | 71.09%   | 43.44% |
|                            | recall      | 41.95%                | 53.11%   | 42.79% | 69.35%                  | 74.47%   | 26.52% |
|                            | specificity | 53.51%                | 56.44%   | 58.93% | 65.74%                  | 67.9%    | 69.52% |
|                            | precision   | 43.47%                | 50.86%   | 48.82% | 67.54%                  | 70.67%   | 26.69% |
|                            | f-score     | 42.56%                | 51.79%   | 44.57% | 68.28%                  | 72.36%   | 26.58% |
| EURUSD                     | accuracy    | 55.39%                | 53.72%   | 53.29% | 66.99%                  | 70.07%   | 46.71% |
|                            | recall      | 40.3%                 | 55.76%   | 56.88% | 57.62%                  | 68.69%   | 47.38% |
|                            | specificity | 67.46%                | 52.17%   | 50.34% | 74.97%                  | 71.24%   | 46.91% |
|                            | precision   | 49.26%                | 47.86%   | 51.43% | 66.15%                  | 66.85%   | 44.93% |
|                            | f-score     | 44.11%                | 51.44%   | 53.89% | 61.50%                  | 67.68%   | 45.92% |

**Table 3.** Accuracy of binary classification for trend predictions

behavior. This knowledge was also considered by application of adjustable tolerance for each of the Fibonacci ratios of the EW's patterns, which leads to higher amount of found patterns, higher coverage of the market data and also higher control of the trading strategy.

The second part of the experiment was focused on application of the impulsive waves for the trend-prediction algorithm. This was compared with randomly selected input waves. The application of complete impulse waves implied higher performance than the input based on incomplete impulse waves.

Although the conditions of the experiment are not entirely the same from previous studies of the trend-predictions, the results are very comparable which makes the Elliott wave theory still attractive and applicable in this field of interest.

## 5 ACKNOWLEDGMENT

This research was conducted within the Students Grant Competition project reg. no. SP2016/175.

## References

1. W. Huang, Y. Nakamori, S.-Y. Wang, Forecasting stock market movement direction with support vector machine, *Computers & Operations Research* 32 (10) (2005) 2513 – 2522, applications of Neural Networks. doi:<http://dx.doi.org/10.1016/j.cor.2004.03.016>.

2. G. S. Atsalakis, K. P. Valavanis, Forecasting stock market short-term trends using a neuro-fuzzy based methodology, *Expert Systems with Applications* 36 (7) (2009) 10696 – 10707. doi:<http://dx.doi.org/10.1016/j.eswa.2009.02.043>.
3. R. Choudhry, K. Garg, A hybrid machine learning system for stock market forecasting, *World Academy of Science, Engineering and Technology* 39 (3) (2008) 315–318.
4. E. Volna, M. Kotyrba, R. Jarusek, Multi-classifier based on Elliott wave's recognition, *Computers & Mathematics with Applications* 66 (2) (2013) 213–225.
5. G. S. Atsalakis, K. P. Valavanis, Surveying stock market forecasting techniques—part ii: Soft computing methods, *Expert Systems with Applications* 36 (3) (2009) 5932–5941.
6. FXDD, Metatrader 1-minute data (2016).  
URL <http://www.fxdd.com/us/en/forex-resources/forex-trading-tools/metatrader-1-minute-data>
7. Elliott, R. N. Nature's law—the secret of the universe. New York: 63 Wall St., R. N. Elliott, 1946, 64 p, *Science Education* 31 (1) (1947) 44–44.
8. A. J. Frost, R. R. Prechter, *Elliott wave principle: key to market behavior*, Elliott Wave International, 2005.
9. M. Dash, A. Patil, An exploratory study of Elliott wave theory in Indian stock markets, Available at SSRN 1412733.
10. G. S. Atsalakis, E. M. Dimitrakakis, C. D. Zopounidis, Elliott wave theory and neuro-fuzzy systems, in *stock market prediction: The wasp system*, *Expert Systems with Applications* 38 (8) (2011) 9196–9206.
11. S. W. Poser, *Applying Elliot Wave theory profitably*, Vol. 169, John Wiley & Sons, 2003.
12. R. R. Prechter, *Trading the Elliott Waves: Winning Strategies for Timing Entry & Exit Moves*, Marketplace Books, 2003.
13. B. Williams, J. Gregory-Williams, *Trading Chaos.*, John Wiley & Sons, 1995.
14. C. Cortes, V. Vapnik, Support-vector networks, *Machine learning* 20 (3) (1995) 273–297.
15. V. Vapnik, *The nature of statistical learning theory*, Springer Science & Business Media, 1995.
16. K.-j. Kim, Financial time series forecasting using support vector machines, *Neuro-computing* 55 (1) (2003) 307–319.
17. S. Mukherjee, E. Osuna, F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, in: *Neural Networks for Signal Processing [1997]* VII. Proceedings of the 1997 IEEE Workshop, IEEE, 1997, pp. 511–520.
18. K.-S. Shin, T. S. Lee, H.-j. Kim, An application of support vector machines in bankruptcy prediction model, *Expert Systems with Applications* 28 (1) (2005) 127–135.
19. L. Breiman, Random forests, *Machine learning* 45 (1) (2001) 5–32.
20. V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, B. P. Feuston, Random forest: a classification and regression tool for compound classification and QSAR modeling, *Journal of chemical information and computer sciences* 43 (6) (2003) 1947–1958.
21. A. Prinzie, D. Van den Poel, Random forests for multiclass classification: Random multinomial logit, *Expert systems with Applications* 34 (3) (2008) 1721–1732.
22. J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
23. R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: *Ijcai*, Vol. 14, 1995, pp. 1137–1145.