Regarding the Selection of a Trading Strategy in Efficient Markets

Arnaldo Vergara-Romero[1], Petr Sed'a[2], César Pozo-Estupiñan[3] and Lisette Garnica-Jarrin[4]

[1]Universidad de Córdoba, Córdoba, España
[2]VSB-Technical University of Ostrava, Ostrava, Czech Republic
[3]Universidad Bolivariana del Ecuador, Durán, Ecuador
[4]Axios Research EC, Daule, Ecuador

Abstract
The effectiveness of information processing in stock markets is frequently subjected to statistical analysis and verification. If a particular stock market is judged to be inefficient, it is reasonable to devise and implement an exchange plan that, if successful, would result in returns that are greater than the market average. Because nonlinear dependence in returns is typically the root cause of information inefficiency in stock markets, it is prudent to employ an exchange plan that is founded on nonlinear relationships to maximize one's chances of financial success. This article's purpose is not to present statistical verification of the effectiveness or ineffectiveness of the marketplace in question; rather, it is to propose the principle underlying this exchange plan and provide an illustrative example based on actual data. The nonlinear dependencies in the return time series are the foundation of the exchange plan suggested in this paper. The k-nearest neighbor technique, also known as the k-NN technique, is used to generate buy or sell signals. This methodology adheres to the same fundamental underpinnings as the nonlinear BDS test. Python, which is a programming language, is going to be used to implement the k-NN technique.

Keywords
Trading Strategy, Efficient Markets, Exchange Strategies, Bootstrapped

1. Introduction

The disposition that an investor has toward risk, the time horizon over which they invest, and the degree to which they have faith in the efficient market hypothesis are typically the deciding factors in how that investor chooses to trade [1]. It is abundantly clear that some participants in the market frequently engage in irrational behavior. Evidence of systematic errors made by some investors has been presented by behavioral financial economists and psychologists. When it comes to choosing a exchange plan, rational investors have two primary options to choose from. Since it is a game of no winners and always the same number of losers, passive management may still be a valid tactic in some circumstances. Obviously, somebody must

ICAIW 2023: Workshops at the 6th International Conference on Applied Informatics 2023, October 26–28, 2023, Guayaquil, Ecuador
z82veroa@uco.es (A. Vergara-Romero); petr.seda@vsb.cz (P. Sed'a); capozoe@ube.edu.ec (C. Pozo-Estupiñan); lisg13@gmail.com (L. Garnica-Jarrin)
0000-0001-8503-3685 (A. Vergara-Romero); 0000-0002-2205-6377 (P. Sed'a); 0000-0001-5974-8003 (C. Pozo-Estupiñan); 0000-0002-2440-3766 (L. Garnica-Jarrin)

hold all the stocks, and while some investors are able to generate returns that are higher than average, other investors have to generate returns that are lower than average. When additional costs associated with active management are considered, it is possible that most investors will not outperform the market average; consequently, passive investors may achieve better results. The use of fundamental and technical analyses is an option if a particular market is inefficient. Of course, it is also possible to apply rules or models that are more complex in nature [2].

[3] cites several empirical studies that all came to the same conclusion: returns are conditionally predictable. In the body of academic literature, linear dependencies in financial time series returns have been identified only very infrequently. Only extremely low and extremely high frequencies are exempt from this rule [3]. It is common practice to assume that the potential for nonlinear dependencies is too complex to be accurately specified. As a result, practitioners focus a lot of their attention on the use of artificial intelligence and data mining techniques that do not require the forecaster to precisely describe the associations that exist in the time series. For this reason, a wide variety of data mining techniques, including neural networks, nearest neighbor analysis, classification trees, random forests, and many others, have been implemented to forecast the future values of financial time series [4].

The plan of an exchange approach that would be established on nonlinear relationships in returns and that would therefore combine data mining techniques with modern programming tools is the practical challenge that needs to be addressed. The purpose of this paper is to propose the principle of an exchange plan that is based on nonlinear dependencies detected in returns by the k nearest neighbor (k-NN) technique using the Python programming language, and to show an illustrative example on a real data sample. The k nearest neighbor (k-NN) technique detects nonlinear dependencies in returns by comparing each return to its nearest neighbor.

The value that is brought to the table by this piece of writing can be broken down into the following two categories:

- The correlation between the results of the nonlinear BDS test and exchange plan will be demonstrated.
- The k-NN technique will be used to demonstrate how the possibility of using the programming language Python to detect signals for selling and buying generated by the technique will also be demonstrated.

The following outline describes how this article is structured. In the introduction, the motivation was broken down, a brief overview of the relevant literature and the existing body of knowledge was presented, and the significance of this paper's contribution to the field was emphasized. A theoretical background of the problem that was investigated is presented in the second chapter. It is briefly described how the efficient market hypothesis (EMH), the adaptive market hypothesis (AMH), and the BDS test are all related to one another. In the third section, an exchange plan that makes use of the outcomes of the BDS statistical test, the programming language Python, and the nearest neighbor technique is described. This plan was developed using the aforementioned tools. In addition to that, you will be provided with an illustrative example that contrasts the outcomes of three different exchange strategies. The conclusion of this paper provides a synopsis of the findings and a discussion of those findings.

2. The ability to predict returns under certain conditions, based on nonlinear dependencies

The effects of statistical tests that investigate whether or not returns are conditionally expectable have a fundamental bearing on the manner in which exchange strategies can be practically applied, regardless of whether the market in question is efficient or inefficient [5]. However, conditional predictability is not statistical proof of the market's efficiency or lack of efficiency. The AMH suggests that both efficiency and inefficiency can shift and develop over the course of one's lifetime. As a means of adjusting to shifting market conditions, market agents frequently turn to heuristics to select appropriate investment opportunities. According to [6], the degree to which market conditions can be predicted shifts over time. This is because market conditions can change. Using statistical tools, one can investigate whether or not the returns on assets are conditionally predictable.

The unpredictable nature of price shifts is one of the empirical hallmarks of markets that are informationally efficient. Because of this, the evolution of prices should not be predictable in markets that are efficient. In the scientific literature, several statistical tests, both linear and nonlinear, have been defined to verify the randomness of return values. According to [7, 8], the BDS test is one of the most powerful nonlinear tests that can be used to examine the first type of random walk hypothesis. The BDS test of independence is a non-parametric technique for evaluating the null hypothesis that the data are unrelated to one another and are distributed in an equal fashion [9, 10]. The concept of the correlation integral will serve as the foundation for the examination. The BDS test considers a diverse range of competing hypotheses. Non-stationarity, chaos, and nonlinear stochastic processes are some examples of the types of non-independent and unequally distributed processes that it can identify [11, 12]. When one possesses a sequence of logarithmic returns rt that is comprised of n observations, one can write the correlation integral for m dimensions as follows:

$$C_{m,n}(\varepsilon) = \frac{2}{(n-m+1)*(n-m)} * \sum_{s=1}^{n-m} \sum_{t=s+1}^{n-m+1} \prod_{j=0}^{m-1} H_\varepsilon(r_s + j, r_t + j) \tag{1}$$

where ε is a small enough preset distance, m is a nesting parameter and H is the Heaviside function, the following formula holds true for this function:

$$H_\varepsilon(r_i, r_j) = \begin{cases} 1 \quad if \quad |r_i - r_j| \le \varepsilon \\ 0 \quad for \quad other \quad cases \end{cases} \tag{2}$$

According to [9, 10], test statistics are defined as follows:

$$W_{m,n}(\varepsilon) = \frac{\sqrt{n-m+1} * [C_{m,n}(\varepsilon) - C_{1,n-m+1}(\varepsilon)^m]}{\hat{\sigma}_{m,n}(\varepsilon)} \sim N(0,1) \tag{3}$$

An approximation of the asymptotic standard deviation is denoted by the notation $\hat{\sigma}_{m,n}(\varepsilon)$.

3. An outline of a potential exchange plan

In light of the findings that were presented by [9], in which it was discovered that returns are rather expectable based on the findings of the nonlinear BDS test, it is reasonable to propose an exchange plan that is based on the existence of nonlinear associations in earnings.

3.1. K-nearest neighbor technique

It would appear that the k-NN technique is an appropriate technique for the generation of buy or sell signals. According to [13, 14], this technique involves selecting geometric segments of historical time series values that are comparable to the segment that came immediately before the observation that is going to be predicted. In point of fact, the k-NN technique selects pertinent earlier observations not according to their position in time but rather on the basis of the levels and geometric trajectories they followed [15, 16]. The k-nearest neighbors technique of prediction can be broken down into several distinct steps. The time series x_t, where $t = 1, ..., n$, needs be transformed into a series of segments of the same length. These segments need to have the form of vectors $xt\ m, \tau$ and they need to contain m sample observations of the original time series at intervals $\tau \in N$:

$$x_t^{m,\tau} = (x_t, x_{t-\tau}, x_{t-(m-1)\tau}) \tag{4}$$

Where denotes the nesting dimension, and is the delay parameter, respectively. In the scientific literature, these m-dimensional vectors are referred to as m-histories, whereas the phase space of the time series is represented by the multidimensional space R^m. The nearness of two m-histories in the segment area of the time series can be characterized as nearby neighbors in the dynamical behavior of two sections in the time series x_t.

The forecast of the time series x_t where $t = 1, ..., n$, is based on the analysis of the historical paths of the vectors around the last vector $x_n^m = (x_n, x_{n-1}, ..., x_{n-(m-1)})$. his analysis was carried out so that the x_t time series could be predicted. In the subsequent step, segments with dynamic behaviors that are similar to those previously identified are identified and used for prediction. In order to derive a prediction for a time series, it is necessary to take into account a total of k m-histories, such as $x_{i1}^m, x_{i2}^m, x_{i3}^m, ..., x_{ik}^m$, which are most comparable to x_n^m. In this way, the prediction can be obtained [17]. It is necessary to identify the nearest k vectors in the phase space of time series R^m in order to locate the nearest neighbors of x_n^m. This is done so that the nearest neighbors of x_n^m can be located. The process of identification involves locating the m-history x_i^m that has the highest serial correlation with the most recent vector, which is denoted by x_n^m. The prediction of the time series x_{n+1}^f using the NN technique is accomplished by applying linear autoregression, the coefficients of which are estimated using the technique of least squares. This allows for accurate forecasting of the time series [18]. This represents a regression across the entirety of k m-histories, including $x_{i1}^m, x_{i2}^m, x_{i3}^m, ..., x_{ik}^m$.

It is critical that the prediction made by the k-NN technique depends not only on the value of the nestling measurement m but also on the quantity of points in the phase space of the time series R^m that are closest to k. It is possible to make use of a number of different heuristic techniques, for instance, in order to determine these two parameters [19]. However, the use of genetic algorithms, which enable the concurrent purpose of the optimum values of m and

k, is the technique that is considered to be the most applicable. Genetic algorithms are a type of adaptive examine and optimization techniques. They have the advantage of being able to evaluate the loss functions associated with the predictor parameters without assuming the continuousness or differentiability of the loss function, which is a common assumption with other types of exploration and optimization techniques [20]. The use of genetic algorithms offers an additional benefit in that it removes the issue of data snooping, which is another advantage of using genetic algorithms. The sample has been separated into a training set and an exam set. However, prior to utilizing the k-NN technique, it is recommended to first determine whether or not the time series of returns contains any nonlinear dependence. The demonstration of nonlinearity would provide conclusive evidence that our line of thought regarding the implementation of the k-NN technique is on the right track. It would appear that the BDS test, which follows a similar line of reasoning, is the one that is most applicable [21, 22].

3.2. K-nearest neighbor technique

Python is a high-level programming language that supports multiple programming paradigms and provides dynamic control over the data types that are used. These programming paradigms include object-oriented programming, imperative programming, and functional programming. Python is an open-source software program that provides free installation packages for the majority of operating systems and computing platforms (Unix, MS Windows, macOS, Android). In the world of applied finance, the programming language Python is also experiencing a meteoric rise in popularity right now.

In point of fact, the k-NN algorithm produces an imaginary boundary for the data classification process. The algorithm makes its predictions relative to the nearest boundary line whenever new data points are added. As a result, having a higher value for the parameter k implies a damping of the separation curves, which ultimately results in models that are simpler. While a smaller value for the k parameter has a tendency to overfit the data and result in complex models, larger values tend to fit the data more accurately. In conclusion, selecting the appropriate value for the k parameter is extremely important if one wishes to avoid either overfitting or underfitting the data set [23]. Python code implementing the nearest neighbor algorithm should proceed through the following series of steps:

- Import the k-NN algorithm that is contained within the scikit-learn package.
- Develop variables for features and objectives.
- Separate the data into the test data set and the training data set.
- Create a k-NN model based on the value of the neighboring node.
- Train or fit data into a model.
- Try to anticipate the values in the future

The most important question is how to compute an appropriate value for the k parameter when working with a data set. To obtain the range of the expected k-value, it is obvious that we need to obtain knowledge of the data sample first. This is obvious even at a cursory glance. It is necessary to test the model for each expected k-value if we are to obtain a k-value that is accurate [24]. Figure 1 displays some code written in Python as an example.

```
# Import necessary modules
gdf from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_SP500return
import numpy as np
import matplotlib.pyplot as plt
SP500returnData = load_SP500return()
# Create feature and target arrays
X = SP500returnData.data
y = SP500returnData.target
# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, ran-
dom_state=42)
neighbors = np.arange(1, 9)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))
# Loop over K values
for i, k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
       # Compute training and test data accuracy
    train_accuracy[i] = knn.score(X_train, y_train)
    test_accuracy[i] = knn.score(X_test, y_test)
```

Figure 1: Python code demonstrating an example of testing the accuracy of the k-nearest neighbor model.

3.3. K-nearest neighbor technique

The exchange plan that has been proposed is founded on the concept of straightforward market timing, which entails either investing in the stock market (an index or asset), or in a risk-free asset. The k-NN technique's forecasts are used to divide the next exchange day into two categories, and each category corresponds to a different exchange option [25]. The first choice involves the investor's participation in the market (which will result in a return proportional to the market), while the second choice involves the investor's withdrawal from the market and an investment in a safe asset (it will bring a riskfree return). The exchange plan outlines the position for the following exchange day by taking into account the current state (whether there is presence or absence in the market) and generating buy or sell signals through the application of the k-NN technique. If, under the circumstances of market presence, it is anticipated that prices will go down based on the prediction made using the k-NN technique, then the asset is sold, and the proceeds are invested in an asset that does not involve any risk [26]. On the other hand, if there is no activity in the market at all, the k-NN technique forecasts a rapid increase in price in the not-too-distant future, and this results in the generation of a signal to buy the asset. After the sale of the risk-free asset, the proceeds are placed in market-based investments. When selecting either of the other two possibilities, the status quo will be maintained [27].

When the exchange rule is applied to the entirety of the exchange period, beginning at time 1 and ending at time T, the yield r, which can be written as follows:

$$r = \sum_{t=1}^{T} r_m(t) * I_b t + \sum_{t=1}^{T} r_f(t) * I_s(t) + n * \log \frac{1-c}{1+c} \tag{5}$$

Where $r_m(t)$ represents the market return, $I_b t$ and $I_s t$ are indicator variables that take on the value of one if the k-NN technique produces a signal for buying or selling, respectively, and take on the value zero in all other circumstances. It is required that the condition $I_b t * I_s t = 0, \forall t \in (1, T)$ be satisfied, where n is the total number of transactions, and the value of c represents the transaction costs expressed as a percentage of the price [28].

If the value of the prediction at time t is close to the value of the closing price at time $t-1$, then it is convenient to modify a simple rule for buying or selling with a filter that reduces the number of false signals for buying or selling [29]. This can be done by checking to see if the prediction at time t is close to the value of the closing price at time $t-1$. One way to think about the filter is as the quantity of risk that the stakeholder is disposed to take on. If the expected value obtained by the k-NN technique is higher (lower) than the closing price at time $t-1$, adjusted by the percentage of the standard deviation of the price differences on the interval from 1 to $t-1$, the filter rule will generate buy (sell) signals at time t. This occurs when the closing price at time $t-1$ is compared to the expected value obtained by the k-NN technique. If, $\hat{P}_t$ is the value that was predicted for the price $\P(t)$, then a buy signal will be generated if, $\hat{P}_t$ is greater than $P_{t-1} + \delta.\ \sigma$ while the investor is absent from the market at the same time [30]. Given that we already have a presence in the market, we shouldn't withdraw from it. On the other hand, a sell signal is generated if the condition, $\hat{P}_t \leq P(t-1) + \delta.\ \sigma$ applies and the investor is present on the market at the same time. In the event that we do not have a presence in the market, we should keep holding onto the risk-free asset [31].

3.4. K-nearest neighbor technique

It is necessary to consider how the k-NN exchange plan proposed stacks up against other possible approaches. In order to keep this piece to a reasonable length, we will only provide one example for illustration purposes. The exchange strategies that are chosen for comparison are going to be based on the Relative Strength Index (RSI) and the Relative Momentum (RM) [32].

In light of the fact that the k-NN technique is predicated on the presence of non-linear relationships in returns, it is recommended that the BDS test be utilized in order to validate the non-linearity of the data sample in question. Asymptotic estimates are used as the foundation for the BDS test statistic. On the other hand, it may cause one to draw incorrect inferences [31]. The bootstrap technique was utilized so that this issue could be resolved.

In this approach, individual and combined statistical tests are computed by making use of samples of T-observations that are produced through the process of weighting the initial data. The fraction of replications in a bootstrapped study that are external the restrictions described by the predictable measurements is what is used to directly calculate the bootstrapped study's p-value. It was decided that there would be 2500 replications of each experiment.

The daily returns of the S&P 500 index were selected for inclusion in this article. In Figure 2, you can see how the p-values of the non-linear BDS test statistics for the S&P 500 index returns changed over the course of the period from 2006 to 2017. Calculations were made to determine the values of BDS statistics for nesting dimensions 2 and 3. (marked by the symbols BDS2 and BDS3). The level of statistical significance that will be used to evaluate the significance of BDS tests is set at 5%, and this level is represented in the graphs by a horizontal line in the color red. On the basis of a time series that spanned twenty months, the p-values were calculated. The p-value for October 2008, for instance, corresponds to the data sample for the time period spanning February 2007 through September 2008. As can be seen in Figure 2, the assumption of complete independence between returns is not supported by the data for the years 2011-2012 and 2014-2015.

As a starting point for the calculations, we used the dividend-adjusted daily closing rates of the S&P 500 index that were in effect during the four-year time span beginning January 1, 2011, and ending December 31, 2014. This time frame was specifically selected on purpose. Figure 2 presents the findings that indicate that the BDS test conducted during that time period frequently led to the rejection of the null hypothesis of independence. The information was gathered from the website located at http://finance.yahoo.com. As a risk-free investment option, a United States Treasury bill with a maturity of three months was chosen. The transaction costs, denoted by c, were set at a level equal to five percent of the total price. Additionally, the percentage of the standard deviation that corresponds to the price differences was established at 5 percent [33].

It is recommended that the 14-day indicator be used for the RSI calculation because it represents the most suitable time period. Therefore, the same frequency was selected for the RM indicator, when typically, a value from the range of 5-25 exchange days is selected [34].

Table 1
Summary results of exchange strategies.

Plan	Profiability	Average yield
k-NN plan	84%	57%
RM 14 plan	69%	48%
RSI 14 plan	86%	53%

The outcomes of all three exchange strategies are compared and summarized in Table 1. The profitability indicator presents the proportion of total trades that resulted in a profit relative to the total number of trades executed. The most important feature of each exchange plan is expressed as a percentage and represents the typical return obtained by using that plan. According to the findings, during the evaluation period, the k-NN plan performed marginally better than the other two options when the average yield was taken into consideration. This is, of course, merely a hypothetical illustration of how one could use historical data samples without actually investing any money.

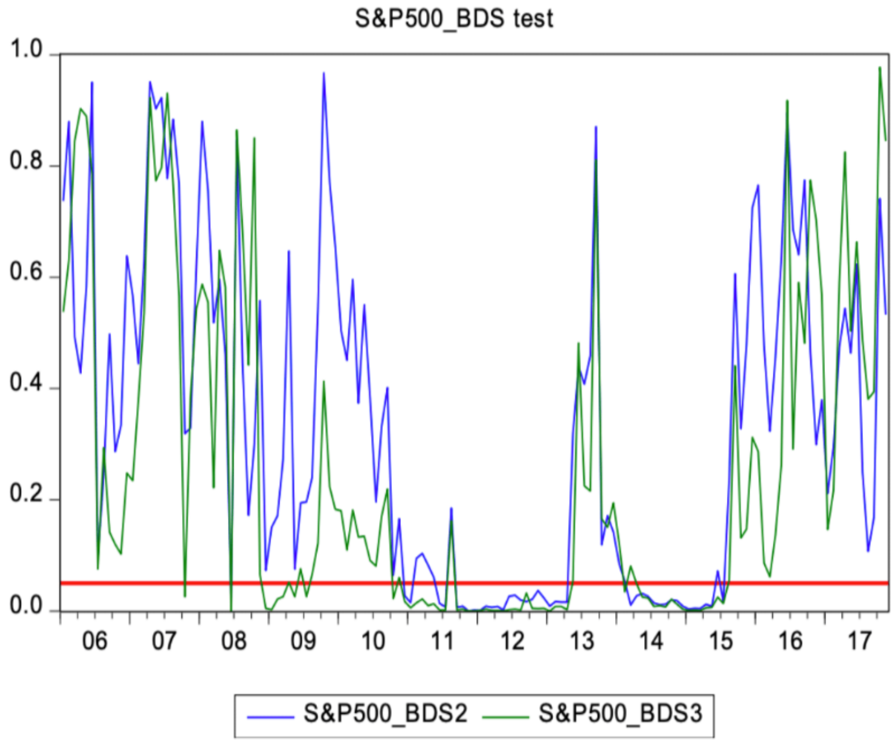

Figure 2: Bootstrapped p-values of the BDS test with nesting dimensions 2 and 3.

4. Conclusions

Within the scope of this research project, a k-NN-based exchange plan was proposed for consideration. However, the purpose of this paper was not to test that approach on existent statistics samples and thus specify statistical verification of the effectiveness or ineffectiveness of the given marketplace. Instead, the purpose of this paper was to discuss the topic at hand. The purpose of this paper was to propose only the principle of an exchange plan that is established on nonlinear dependencies detected in revenues, utilize the Python programming language, and show an illustrative example on a real data sample. Specifically, the goal was to show that nonlinear dependencies can be detected in returns. The nonlinear k-NN technique is responsible for the generation of buy or sell signals for the proposed exchange plan.

The nonlinear BDS independence test and the k-NN technique are both based on similar underlying principles, specifically the correlation integral and nesting dimensions. As a result, there is an unmistakable link between the findings of the statistical analysis and the exchange plan that has been suggested. In addition, the article demonstrates the direct utilization of the k-NN technique in the Python programming environment, which is a tool that is becoming increasingly popular not only in empirical finance but across the board. The profitability of the suggested exchange plan was evaluated using real data derived from the US stock market during the period 2011-2014, which was a time when the results of the BDS test frequently confirmed non-linear dependencies in return values. After that, the k-NN plan was compared to exchange strategies based on RSI and RM to see which one performed better.

The empirical findings lean somewhat in favor of a tactic that is founded on the k-NN technique. In addition to testing the proposed k-NN plan on data from other markets, for a variety of assets, at various stages of market development, for a variety of data frequencies, in subperiods of varying lengths, and comparing the results of these tests to those of other types of strategies, the proposed plan should also be tested on data from other markets. It is also recommended that a statistical investigation be carried out to determine whether the profitability of the k-NN plan is in any way connected to the outcomes of the BDS test.

References

[1] A. Urquhart, F. McGroarty, Are stock markets really efficient? evidence of the adaptive market hypothesis, International Review of Financial Analysis 47 (2016) 39–49.

[2] A. R. Pagan, K. A. Sossounov, A simple framework for analysing bull and bear markets, Journal of applied econometrics 18 (2003) 23–46.

[3] P. F. Christoffersen, F. X. Diebold, Financial asset returns, direction-of-change forecasting, and volatility dynamics, Management Science 52 (2006) 1273–1287.

[4] R. C. Cavalcante, R. C. Brasileiro, V. L. Souza, J. P. Nobrega, A. L. Oliveira, Computational intelligence and financial markets: A survey and future directions, Expert Systems with Applications 55 (2016) 194–211.

[5] A. Morante, M. Villamil, H. Florez, Framework for supporting the creation of marketing strategies, Information 20 (2017) 7371–7378.

[6] A. W. Lo, The adaptive markets hypothesis: Market efficiency from an evolutionary perspective, Journal of Portfolio Management, Forthcoming (2004).

[7] P. Sed'a, Teoretické aspekty hypotézy efektivního trhu a empirie na akciových trzích v zemích střední evropy, 2020.

[8] J. F. Romero-Subia, J. A. Jimber-del Rio, M. S. Ochoa-Rico, A. Vergara-Romero, Analysis of citizen satisfaction in municipal services, Economies 10 (2022) 225.

[9] W. A. Broock, J. A. Scheinkman, W. D. Dechert, B. LeBaron, A test for independence based on the correlation dimension, Econometric reviews 15 (1996) 197–235.

[10] A. Vergara-Romero, J.-A. Jimber-del Río, F. Márquez-Sánchez, Food autonomy within food sovereignty: Evidence from a structural model, Agronomy 12 (2022) 1141.

[11] P. Sed'a, L. Caridad Lopez del Rio, Conditional predictability of assets returns: a dynamic approach, 2021.

[12] M.-S. Ochoa-Rico, J.-A. Jimber-del Río, G. Cornejo-Marcos, A. Vergara-Romero, Characterization of the territory and estimation of a synthetic index of social welfare., TEM Journal 11 (2022).

[13] F. Fernandez-Rodrıguez, S. Sosvilla-Rivero, J. Andrada-Felix, Exchange-rate forecasts with simultaneous nearest-neighbour methods: evidence from the ems, International Journal of forecasting 15 (1999) 383–392.

[14] G. Macas-Acosta, G. Macas-Lituma, A. Vergara-Romero, The internal and external factors that determined private investment in ecuador 2007–2020, Economies 10 (2022) 248.

[15] G. Lin, A. Lin, D. Gu, Using support vector regression and k-nearest neighbors for short-

term traffic flow prediction based on maximal information coefficient, Information Sciences 608 (2022) 517–531.
[16] D. Chumachenko, I. Meniailov, K. Bazilevych, T. Chumachenko, S. Yakovlev, Investigation of statistical machine learning models for covid-19 epidemic process simulation: Random forest, k-nearest neighbors, gradient boosting, Computation 10 (2022) 86.
[17] L. Sun, J. Zhang, W. Ding, J. Xu, Feature reduction for imbalanced data classification using similarity-based feature clustering with adaptive weighted k-nearest neighbors, Information Sciences 593 (2022) 591–613.
[18] S. Bouzebda, A. Laksaci, M. Mohammedi, The k-nearest neighbors method in single index regression model for functional quasi-associated time series data, Revista Matemática Complutense 36 (2023) 361–391.
[19] M. Mailagaha Kumbure, P. Luukka, A generalized fuzzy k-nearest neighbor regression model based on minkowski distance, Granular Computing 7 (2022) 657–671.
[20] J. H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, 1992.
[21] D. S. Tripathy, B. R. Prusty, K. Bingi, A k-nearest neighbor-based averaging model for probabilistic pv generation forecasting, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 35 (2022) e2983.
[22] M. Cubillos, S. Wøhlk, J. N. Wulff, A bi-objective k-nearest-neighbors-based imputation method for multilevel data, Expert Systems with Applications 204 (2022) 117298.
[23] D. Sisodia, D. S. Sisodia, Quad division prototype selection-based k-nearest neighbor classifier for click fraud detection from highly skewed user click dataset, Engineering Science and Technology, an International Journal 28 (2022) 101011.
[24] M. Hu, E. C. Tsang, Y. Guo, D. Chen, W. Xu, Attribute reduction based on overlap degree and k-nearest-neighbor rough sets in decision information systems, Information Sciences 584 (2022) 301–324.
[25] K. Yuan, D. Yu, J. Feng, L. Yang, C. Jia, Y. Huang, A block cipher algorithm identification scheme based on hybrid k-nearest neighbor and random forest algorithm, PeerJ Computer Science 8 (2022) e1110.
[26] Z. Gao, Y. Lin, X. Sun, X. Zeng, A reduced order method for nonlinear parameterized partial differential equations using dynamic mode decomposition coupled with k-nearest-neighbors regression, Journal of Computational Physics 452 (2022) 110907.
[27] J. Wang, Z. Zhou, Z. Li, S. Du, A novel fault detection scheme based on mutual k-nearest neighbor method: Application on the industrial processes with outliers, Processes 10 (2022) 497.
[28] A. A. Nababan, M. Khairi, B. S. Harahap, Implementation of k-nearest neighbors (knn) algorithm in classification of data water quality, Jurnal Mantik 6 (2022) 30–35.
[29] C. Zhang, P. Zhong, M. Liu, Q. Song, Z. Liang, X. Wang, Hybrid metric k-nearest neighbor algorithm and applications, Mathematical Problems in Engineering 2022 (2022) 1–15.
[30] L. M. Elshenawy, C. Chakour, T. A. Mahmoud, Fault detection and diagnosis strategy based on k-nearest neighbors and fuzzy c-means clustering algorithm for industrial processes, Journal of the Franklin Institute 359 (2022) 7115–7139.
[31] A. S. Abdulameer, S. Tiun, N. S. Sani, M. Ayob, A. Y. Taha, Enhanced clustering models with wiki-based k-nearest neighbors-based representation for web search result clustering,

Journal of King Saud University-Computer and Information Sciences 34 (2022) 840–850.
[32] R. D. Edwards, J. Magee, W. C. Bassetti, Technical analysis of stock trends, CRC press, 2018.
[33] K. Georgiev, Y. S. Kim, S. Stoyanov, Periodic portfolio revision with transaction costs, Mathematical Methods of Operations Research 81 (2015) 337–359.
[34] J. Veselá, Investování na kapitálových trzích, 2019.