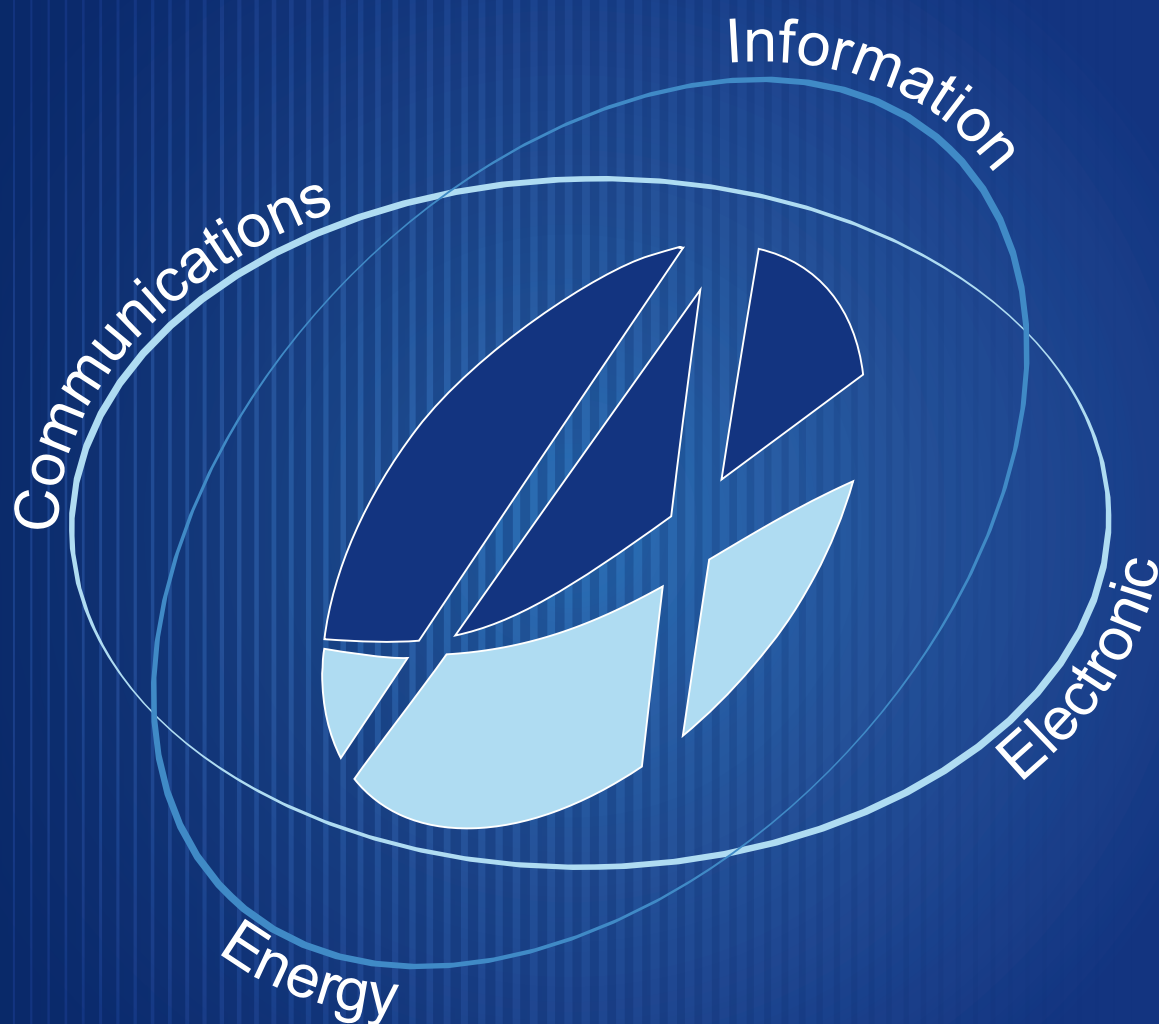


The Journal of **CIEES**

Communications, Information, Electronic & Energy Systems

Print ISSN: 2738-7283
Online ISSN: 2738-7291

Year 2022
Volume 2 (2)



No liability is assumed by the editorial staff for the content of scientific papers and opinions published in this volume. They represent the author's point of view.

Editorial

As a result of widening the spectrum of communication, dissemination and implementation of scientific research results, in 2020 was founded the association journal (magazine), issued under the name The Journal of CIEES appearing both in print (ISSN: 2738-7283), and online (ISSN: 2738-7291) format with DOI: <https://doi.org/10.48149/jciees>.

The Journal of CIEES is published in English and it is open access, peer-reviewed international scientific journal publishing papers submitted by authors themselves for publishing and invited by the Editorial Board. The Journal of CIEES covers all the essential branches of theories, technologies, systems, tools, applications, work in progress, and experiences on all theoretical and practical issues arising in telecommunications, electronics, signal processing and applications, information technologies, energy systems and etc.

As always, many people have contributed to this issue including all authors, the anonymous reviewers, the Associate Editors, the Editorial Advisory Board and, of course, our outstanding production team. We wish to acknowledge each individual's contributions, and to also express our gratitude for everyone's commitment. We hope you enjoy this first issue of The Journal of CIEES.

We have had the privilege to publish truly outstanding, quality manuscripts that, judging by the response of our reviewers and readers, are important contributions to their respective fields of research. We want to take this moment to thank all of our reviewers and authors for entrusting us with their discoveries.

Editors-in-Chief:

Prof. TEODOR ILIEV, PhD
University of Ruse, Bulgaria

&

Prof. IVAYLO STOYANOV, PhD
University of Ruse, Bulgaria

MANAGING EDITORIAL BOARD

Editors-in-Chief:

Prof. Teodor ILIEV, PhD
University of Ruse
Bulgaria

Prof. Ivaylo STOYANOV, PhD
University of Ruse
Bulgaria

Associate Editor-in-Chief:

Prof. Grigor MIHAYLOV, PhD
University of Telecommunications and Post
Bulgaria

Chairman of the Editorial Advisory Board:

Prof. RENATO FILJAR, PhD
Krapina University of Applied Sciences
Croatia

Editorial Advisory Board:

Prof. Iliya ILIEV, PhD - University of Ruse, Bulgaria
Assoc. Prof. Bogdan POPA, PhD - University POLITEHNICA of Bucharest, Romania
Assoc. Prof. Gani BALBAYEV, PhD - Academy of logistics and transport, Kazakhstan
Prof. Felipe PENARANDA-FOIX, PhD - Universidad Polit cnica de Valencia, Spain
Assoc. Prof. Angel TERZIEV, PhD - Technical university of Sofia, Bulgaria
Prof. Goran DJORDJEVIC, PhD - University of Nis, Serbia
Assoc. Prof. Selahattin KOSUNALP, PhD - Bandirma Onyedi Eylul University, Turkey
Assoc. Prof. Stanimir SADINOV, PhD - Technical university of Gabrovo, Bulgaria
Assoc. Prof. Marko HORVAT, PhD - University of Zagreb, Croatia
Assist. Prof. Danijela EFNUSHEVA, PhD - University "Ss. Cyril and Methodius", Republic of North Macedonia
Lecturer Irena POPOVIC - Krapina University of Applied Sciences, Croatia

Web Design Editor:

Assoc. Prof. Yordan DOYCHINOV, PhD
University of Ruse
Bulgaria

English Language Editor:

Assoc. Prof. Diana ILIEVA, PhD
University of Telecommunications and Post
Bulgaria

The Journal of CIEES

Volume 2, Issue 2 / 2022

Print (ISSN: 2738-7283) & Online (ISSN: 2738-7291)

<http://journal.ciees.eu>

E-mail: publication@ciees.eu

Published by: CIEES Association Press, Bulgaria

Copyright: Association on Communications, Information, Electronic and Energy Systems

CONTENTS

Spatial Machine Learning Personal Mobility Predictive Model Trained with Smartphone-Collected Trajectory Data.....	7
Boyan Stoyanovski, Teodor B. Iliev, Radovan Cesarec and Renato Filjar	
Multi Repository Management Tools.....	13
Ulvi Shakikhanli and Vilmos Bilicki	
Solar Panel Tracker System Using GPS Technology	19
Alina Fazylova, Anuar Kussainov, Akzhaina Baidullina and Asel Smailova	
Statistical Analysis of Counter-Covid-19 Policies Adequacy and Effectiveness in Croatia.....	25
Filip Šklebar, Marko Mikša and Robert Spudic	
Towards an Approach to Improve an Environmental Management System in Industrial Organization.....	32
Neli Babekova	
Trajectory Analysis of Griffon Vulture in Kvarner Bay	36
Stella Dumenčić, Lucija Žužić and Barbara Breš	

Spatial Machine Learning Personal Mobility Predictive Model Trained with Smartphone-Collected Trajectory Data

Boyan Stoyanovski ^{1*}, Teodor B. Iliev ¹, Radovan Cesarec ² and Renato Filjar ³

¹ University of Ruse/Department of Telecommunications, Ruse, Bulgaria

² Krapina University of Applied Sciences, Krapina, Croatia

³ Krapina University of Applied Sciences/Laboratory for Spatial Intelligence, Krapina, Croatia

*E-mail of corresponding author: stoyanovskiboyan@gmail.com

Abstract: Individual and group mobility is an essential information for numerous segments of technology (including transport and logistics), society, and economy. The ability of telecommunications devices, such as smartphones, to collect accurate and reliable data on personal mobility with the embedded sensors, inspires research in personal mobility. We confirm the ability of suitably defined indicators to compare sets of trajectories, and identify outliers/differences among the individual ones. Furthermore, we demonstrate development of a machine learning (ML) regression predictive model based on experimental data collected on the real urban environment of the city of Krapina, Croatia, suitable for utilisation in personal mobility analysis, and traffic and transport planning and optimisation.

Keywords: GNSS, trajectory, kinematics, machine learning, predictive model, mobility.

Received 11 October 2022

Accepted 10 November 2022

Published 19 December 2022

1. Introduction

Individual and group mobility is an essential information for numerous segments of technology (including transport and logistics), society, and economy. Different methods for mobility analysis and estimation exist, based on a wide range methods for different data collection. The ability of telecommunications devices, such as smartphones, to collect accurate and reliable data on personal mobility with the embedded sensors, inspires research in personal mobility. A Rich set of mobility related sensors, including a GNSS receiver and accelerometers, embedded in modern smartphones render a fundamental framework in support of the telecommunications-related Location-Based Services, but may be exploited in a range of other applications.

The fact leads to attempt to analyse data in a systematic manner, using methods of trajectory analysis [1] deployed as a bespoke software developed for the purpose of this research using the special library trajr [2] in the R environment for statistical computing [3] using its RStudio Graphical User Interface (GUI) [4]. We attempted to confirm the hypothesis of ability of suitably defined indicators to compare sets of trajectories, and identify outliers/differences among the individual ones. Furthermore, we aimed at showing that the analysis of multiple trajectories on similar paths in a given transport/traffic network may yield the objective indices of variance in trajectories, thus justifying consideration of their utilisation for traffic and transport planning and optimisation. In practical demonstration of the

assumption, the trajectory analysis presented allows us to develop a machine learning (ML) regression predictive model based on experimental data collected on the real urban environment of the city of Krapina, Croatia.

The research presented comprises tasks as follows: Global Navigation Satellite System (GNSS) position estimate collection in scenario of individual mobility, data cleaning and collation, aggregation of GNSS position estimations into trajectories, spatial data analysis and mobility analysis, graphical presentation and interpretation of the results. Methodology, research results, and discussion, resulting from this research, are presented in the rest of this document.

2. Material and research scenario

GNSS position estimates, taken by a GNSS receiver embedded in an Android smartphone, served as a raw material. In the period between 6 July, 2022 to 20 August, 2022, with 119 records in total, a path was walked by one of us (BS) between the place in Dolac Street, Krapina, Croatia and the Krapina University of Applied Sciences (KUAS) building, through the city centre of Krapina, Croatia, as shown in Figure 1. Figure 1 was prepared using a tailored programme developed for the purpose of this research in the open-source R environment for statistical computing [5], and utilised spatial data from the open-access OpenStreetMap database, and the R-library called leaflet. Slight diversions and variations of the path walked were introduced intentionally to allow for variances in the trajectories.

The AndroSensor, an Android-based application, was used to collect GNSS position estimates twice every second (sampling period of 0.5 s).



© The Author(s). Published by CIEES Association Press. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 Generic License.

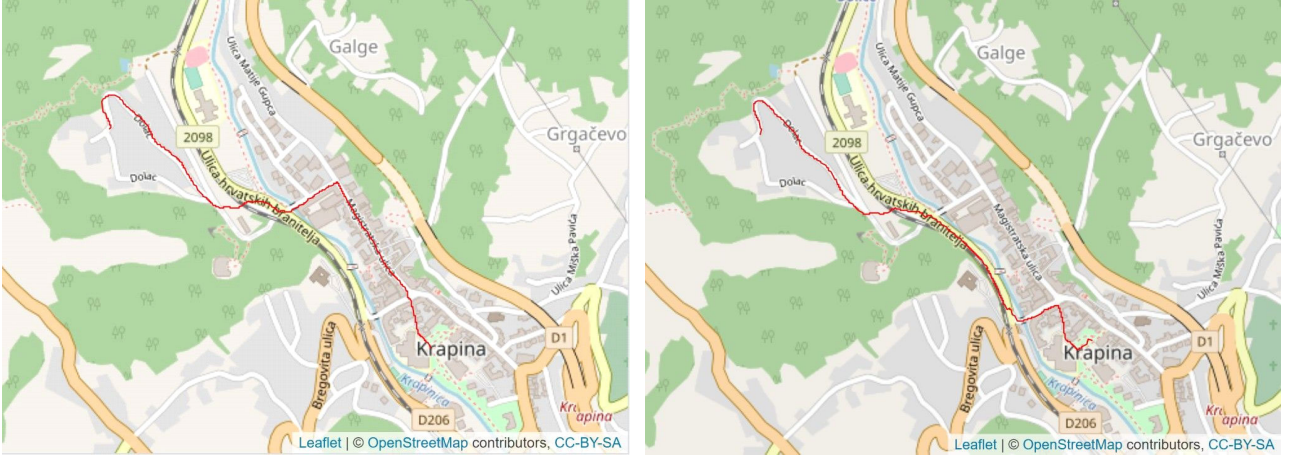


Fig. 1. Two individual paths commonly taken during research, from the starting point at Dolac Street, Krapina to the end-point of KUAS facility in the Krapina city centre. The red line represents a trajectory assembled from a set of GNSS position estimates as a spatial trajectory object, in the R environment for statistical computing.

A set of GNSS position estimates collected during every round of walk, was used to define a trajectory. AndroSensor reads and stores readings of other smartphone sensors, such as accelerometers and gyros, but those have not been used in this research, but remain in the experimental dataset collected to be exploited in future ones.

Collected data are examined for quality, samples with rare missing observations (NA) are removed, and data collated into a machine-readable format. Collected, aggregated, and collated data sets are stored in a CSV database.

3. Methodology for trajectory analysis and modelling

Formal description of trajectory analysis was performed based on the statistical methodology outlined in related references [1] and [2]. The following indices describing individual trajectories were used in this research as trajectory indices (features, statistical variables describing a trajectory): (1) (diffusion) distance of points (position estimates) on individual trajectory; (2) trajectory straightness; (3) trajectory fractal dimension; (4) mean travel velocity, and (5) duration of the path travelled for an individual trajectory.

Indices are defined as follows.

(Diffusion) distance is defined in terms of connectivity, or similarity between two trajectory points x_i and x_j , observed with the time separation of t , and belonging to the data set X as a connected graph [5]. A measure space (X, A, μ) is to be defined on X , with A denoting a set of sub-sets of X , and μ denoting a measure on (X, A) . Let us express the connectivity k between two arbitrary data points of X , x and y , in terms of definition of probability $p(x, y)$ of travelling from x to y in a single random walk step. The probability may be expressed using a suitable kernel function, a Gaussian kernel being a function used in this research, as given in (1), with ε as a parameter.

$$k(x, y) = \exp\left[-\frac{x-y}{\varepsilon}\right] \quad (1)$$

With (X, k) defined as above, a reversible discrete time Markov chain may be developed on X , as given in (2).

$$d(x) = \int_X k(x, y) d\mu(y) \quad (2)$$

Finally, the probability of a single-step random walk may be determined as given in (3).

$$p(x, y) = \frac{k(x, y)}{d(x)} \quad (3)$$

The formal expression of diffusion distance may be then defined with (4) and (5).

$$D_i(x_i, x_j)^2 = \sum_y \frac{((p(y, t|x_i)) - (p(y, t|x_j)))^2}{\phi_0(y)} \quad (4)$$

$$\phi_0(y) = d \frac{(y)}{\sum_{z \in X} d(z)} \quad (5)$$

A (diffusion) distance index value is determined in this research based on the method (4) & (5) deployment in the *trajr* R library [6], and assigned to every trajectory of the 119 element set.

Straightness of a trajectory is considered as a measure of efficiency of a direct walk [2], [6], and defines as a ratio between the shortest (Euclidean) distance between the start and the end points, D , and the actual length of a trajectory, L , calculated as a summation of distances of neighbouring position samples on a trajectory. Straightness index is determined as given in (6) [2], [6].

$$S = \frac{D}{L} \quad (6)$$

A straightness index value is determined in this research based on the method (6) deployment in the *trajr*

R library, and assigned to every trajectory of the 119 element set.

Fractal dimension of a trajectory is regarded as a measure of a geometrical smoothness of the trajectory concerned. It allows a researcher to get a better insight in effects of local topography on trajectory development, without the need for a formal description of the environment surrounding a walker. Estimation of the fractal dimension index is based on the box-counting method proposed by B Mandelbrot in [7], and further refined for ecological/trajectory research by in [8]. In this research, a fractal dimension index value is determined based on the method [8] deployed in the *trajr* R library [6], and assigned to every trajectory of the 119 element set.

Mean travel velocity extends an average value of velocity along the trajectory concerned. The *trajr* R library determines the mean travel velocity for a given trajectory as a vector [6]. This research utilises the length (absolute value) of this vector as a descriptive trajectory index.

Duration of the path travelled for an individual trajectory is measured directly for every trajectory as the time difference between timestamps of end point (last position estimate in trajectory) and starting point (first position estimate in trajectory), respectively.

Trajectory is described by its points, and the values of the above-stated indices. It should be noted that the information on the trajectory context (environment) is embedded indirectly within the set of indices. Multiple trajectory analysis is performed using the trajectory indices that were analysed using exploratory statistical methods applied on sets of trajectory feature values, as statistical variables [9].

This research suggests that the information content of trajectory indices allows for inference in relation to trajectories. This is justified with the demonstration of the application of machine learning methods for prediction of duration of the path travelled (one of the indices, serving as a target statistical variable) based on the knowledge of values of the other indices (those becoming predictors) for the specified trajectory. In the sense of supervised machine learning, several methods for regression predictive model development are applied on the Krapina experimental data set, with the aim of development of the predictive model that returns duration of the path travelled based on known values of diffusion distance, straightness, fractal dimension, and mean travel velocity, which characterise the manner the path was walked. The machine learning model development methods utilised in this research are, as follows: (1) k-Nearest Neighbours (KNN), (2) Support Vector Machine (SVM), (3) Classification And Regression decision Tree (CART), and (4) Random Forest (RF). Detailed outline of machine learning methods utilised may be found elsewhere in excellent references, including [10], and [11]. Particularly, Ceja in [11] outlined the foundations for machine learning in trajectory analysis and prediction, enhancing the motivation for the research presented. The four machine learning methods utilised will be described here comprehensively.

The k-Nearest Neighbours (KNN) algorithm is a simple non-parametric machine learning method that develops regression prediction model based on classification of samples using a reference distance metrics, and assignment of an average value of the target variable in the class to which the predicted sample belongs.

The Support Vector Machine (SVM) method works in a similar manner to the KNN method, as it first classifies samples into classes, and determine the average target variable value of the class to which the predicted sample belongs. However, the means of classification is different from the KNN method, as SVM clusters data using a distance metrics, and then determines the separation plane between the groups. Using the kernel approach (1), the separation plane may be applied to non-linear cases.

The CART method is essentially a decision tree method. It first analyses 'vagueness' in values of variables, and determines the order of appearance of variables along the structure of decision tree. The CART model returns one of the selected outcomes (target variable values) determined during the model development process based on experimental observations.

Finally, the Random Forest (RF) method develops the predictive model by training a large number of decision trees, say 1000, every single one using a different sub-set of original data (observations of statistical variables). Faced with the new case (sample), decision trees will respond with their decisions, and their average is returned as the outcome of the Random Forest model. Compared with the other methods, the RF returns more robust model, which is very accurate, and not prone to over-fitting (modelling noise, rather than signal).

This research utilises deployment of the above-stated predictive model development methods in the R environment for statistical computing, and in particular the specific *caret* R library [12].

Machine learning methods return models in an automated manner. Naturally, the process requires a methodical assessment of performances of the developed models, and selection of one that predicts in the optimal manner [13]. Model performance assessment is performed using a separate data set, not used in the model development procedure. This research splits the original set of statistical variables values into a *training set*, comprising randomly selected samples of amounting to 80% of the *original set*, and a testing set, comprising the rest. The 80-20 split was originally proposed by Pareto [10].

The *caret* R package returns not only the model of the desired structure, but a set of performance indicators that allows of the model performance assessment and comparison with the other models. A good model generally balances its ability to model bias and variance in the phenomenon described by observations. In that terms, this research applies three essential criteria (performance indicators) to select the optimal prediction model for the purpose of prediction of duration of trajectory based on (diffusion) distance of points (position estimates) on individual trajectory, trajectory straightness, trajectory fractal dimension, and mean travel velocity. Performance indicators are derived from the so-called

residual analysis, during which the differences between predicted target values obtained from the model $x_i^{(p)}$, and actually observed target values for the same predictor values x_i , are determined as prediction errors (residuals). Two performance indices describe the success in description of bias: the Mean Absolute Error (MAE), and the Root Mean Square Error (RMSE).

The MAE is determined using (7) on the N samples in the testing sub-set.

$$MAE = \frac{\sum_i |x_i^{(p)} - x_i|}{N} \quad (7)$$

In the manner similar to MAE, the RMSE is defined using (8).

$$RMSE = \sqrt{\frac{\sum_i (x_i^{(p)} - x_i)^2}{N}} \quad (8)$$

An accurate model extends small values of both MAE and RMSE.

The *adjusted coefficient of determination* ($adjR2$) is a widely accepted performance index that describes the model's ability to follow the original variance in observations. The $adjR2$ index is derived from its parent *coefficient of determination* $R2$, in the manner, as follows. The $R2$ index is defined by (9), with variables denoted as in (7), and with $\bar{x}$ denoting the arithmetic mean of x_i .

$$R2 = \frac{\sum_i (x_i^{(p)} - \bar{x})^2 - \sum_i (x_i^{(p)} - x_i)^2}{\sum_i (x_i^{(p)} - \bar{x})^2} \quad (9)$$

The $R2$ index is dependent on the number of samples used for the assessment and the number of predictors in the model. The shortcoming prevents the comparison of models with different number of predictors. Comparison is viable using the $adjR2$ index, defined as in (10), with R^2 determined by (9), n denoting number of samples in dataset, and k denoting the number of independent predictors of the model.

$$adjR2 = 1 - \frac{(1 - R^2)(n - 1)}{(n - k - 1)} \quad (10)$$

Adjusted coefficient of determination may be interpreted as the proportion of the original variance in phenomenon/observations described by the model (for instance, $adjR2=0.92$ means 92% of the original variance is described by the model).

4. Research results

This research exploits observations described in Section 2, using the methodology and the bespoke R software tools described in Section 3.

First, the comprehensive results of statistical analysis of collected data are presented, with utilisation of five statistical variables presenting trajectory indices: (1) (diffusion) distance of points (position estimates) on individual trajectory; (2) trajectory straightness; (3)

trajectory fractal dimension; (4) mean travel velocity, and (5) duration of the path travelled for an individual trajectory, with their scalar values assigned for every of 119 trajectories under consideration.

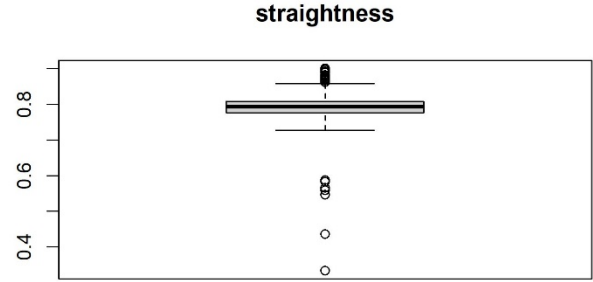


Fig. 2. Box-plot of straightness index values for 119 trajectories.

Figure 2 depicts the box-plot of straightness index values for 119 trajectories.

The box-plot reveals a close similarity in straightness for majority of trajectories, as well as several well-defined outliers. Particular outliers may be easily identified by a slight expansion of the R script.

Figure 3 depicts the box-plot of (diffusion) distance index for 119 trajectories.

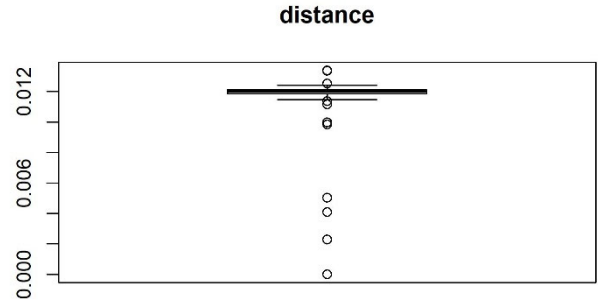


Fig. 3. Box-plot of distance index for 119 trajectories.

It shows even more emphasised unanimity of a majority of trajectories, and even more pronounced outliers.

Figure 4 depicts the box-plot of duration index for 119 trajectories

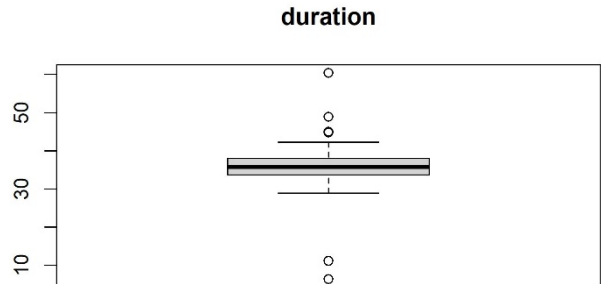


Fig. 4. Box-plot of duration index for 119 trajectories.

Times needed for performing the trajectory pass differ. Still, the majority takes very similar time, while the outliers, comprising significantly different paths, or additions to the common one, are easily identified.

Figure 5 depicts the box-plot of fractal dimension index of 119 trajectories.

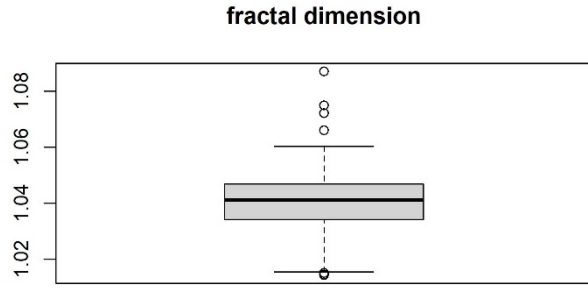


Fig. 5. Box-plot of fractal dimension index of 119 trajectories.

Fractal dimension extends the level of broken lines consisting the trajectory under observation. Although the box-plot extends several outliers, the set examined seems to extend smoothness.

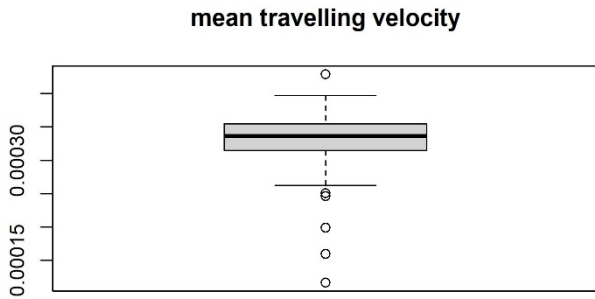


Fig. 6. Box-plot of mean travelling velocity index, across the trajectory concerned, of 119 trajectories.

Mean travelling velocity extends similarity, while still several outliers exist.

The process of regression model development requires the examination of linear association between statistical variables. The correlogram of the variables involved is presented in Figure 7.

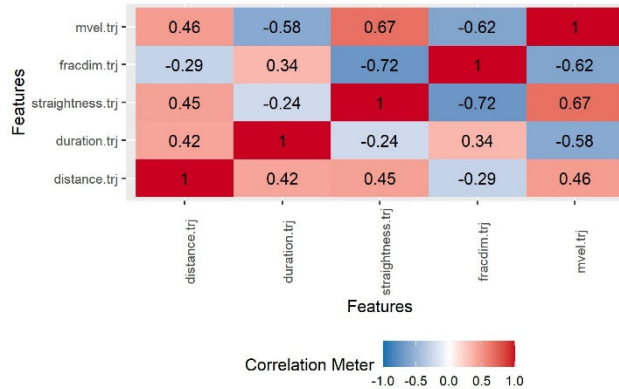


Fig. 7. Correlogram of statistical variables (features: predictors and target) concerned.

The correlogram depicts a general lack of correlation (linear association) both between predictors (in support of linear regression model development), and target and individual predictors (an obstacle for linear regression model development).

The relation between target (duration) and individual predictors is more visible in the segmented boxplots panel, depicted in Figure 8.

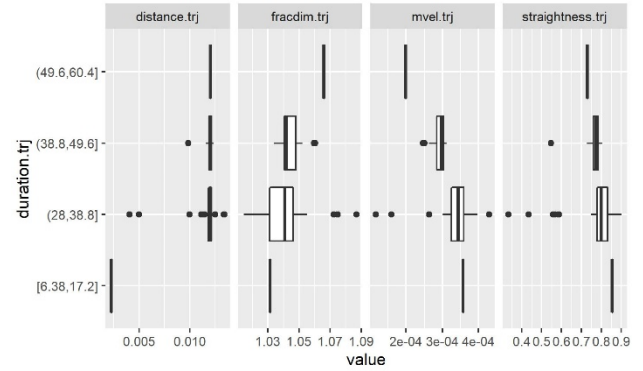


Fig. 8. Panel of box-plot diagrams with segmentation rendered in relation to target (predicted) variable *duration.trj*.

Although diffusion distance does not extend a considerable alignment of box-plots with the ranges of target values, alignment is visible with the other statistical variables assumed as predictors.

The four machine learning predictive model methods: k-Nearest Neighbours (KNN), Support Vector Machine (SVM), Classification And Regression decision Tree (CART), and Random Forest (RF), are applied to yield four predictive models. Performance of the four models are assessed by three criteria: MAE, RMSE, and adjR2, as outlined in Section 3. The performance assessment results are summarised graphically in Figure 9.

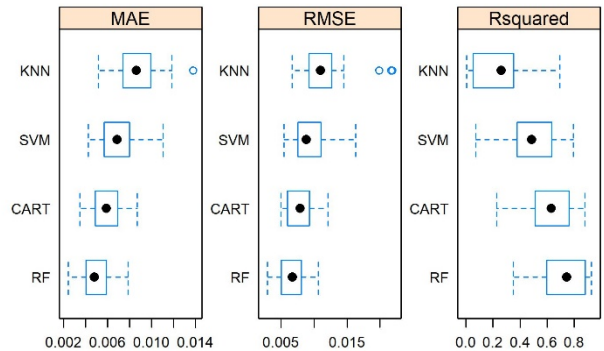


Fig. 9. Performance indices Mean Absolute Error (MAE), Root-Mean Square Error (RMSE), and Rsquared for candidate predictive models developed using k-Nearest Neighbours (KNN), Support Vector Machine (SVM), Classification And Regression decision Tree (CART), and Random Forest (RF) machine learning methods.

Model performance assessment is conducted using the cross-validation approach [11], which yields a set of performance indices values related to different testing sub-sets used. Figure 9 depicts box-plots of such sets. The performance analysis results returns the Random Forest model as the optimal predictive model for the problem of trajectory duration prediction, given the predictor variables values. The RF model evidently extends the smallest median MAE and RMSE, while at the same time covering the largest amount of variance, with median of approx. 75%.

5. Discussion and contribution.

This research aimed at assessment of the ability of several trajectory indices to serve as machine learning descriptors of mobility in the sense of trajectory

classification. We have shown the scalar geometry-based indices describe in trajectory accurately and robustly, while allowing for prediction process without the need for a local topography. The finding renders them possible predictor candidates for machine learning trajectory classification process.

This research assumed the GNSS-based position estimation as flawless, thus presuming the variance in trajectory indices (descriptors) being caused by small intentional variations in path travelled. However, variances in trajectories may result in significant errors in GNSS position estimation caused by non-accounted for effects of space weather/ionospheric disturbances/storms, multipath, or deliberate cyberattacks (spoofing or jamming).

This research will continue with assessment of GNSS position estimation error effects on trajectory identification and classification, and the risk assessment of potential misclassifications.

Statement of open access to r software and data

Authors have made openly available the collection of observations and the developed R-based software to interested scientists and researchers. Citation & access (after 1st December, 2022):

Stoyanovski, B., Iliev, T., Cesarec, R., Filjar, R. (2022). Supplementary material (data & R script) for the manuscript authored by B Stoyanovski, T Iliev, R Cesarec, R Filjar. figshare. Dataset. doi: <https://doi.org/10.6084/m9.figshare.21205304.v2>

Statement of contribution

BS and RF conceived and designed study. BS and RC designed scenario and observation collection methodology. BS collected observations. BS, RF, and TI designed methodology for trajectory analysis and predictive modelling. BS, RF, and TI developed the R-based software. BS and TI analysed trajectories, developed and assessed predictive model. All authors discussed results and drew the conclusion.

References

- [1] Zheng, Y. (2015). Trajectory Data Mining: An Overview. *ACM Transactions on Intelligent Systems and Technology*, 61(3), 1–41. doi: 10.1145/2743025.
- [2] McLean, D J, and Skowron Volponi, M A. (2018). trajr: An R package for characterisation of animal trajectories. *Ethology*, 124, 440–448. doi: 10.1111/eth.12739.
- [3] R project team. (2022). *R version 4.2.1*. Available at: <https://cloud.r-project.org/>.
- [4] RStudio. (2022). *RStudio version 2022.07.0 Build 548*. Available at: <https://www.rstudio.com/products/RStudio/#Desktop>.
- [5] Nadler, B, Lafon, S, Coifman, R R, and Kevrekidis, I G. (2005). *Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck operators*. arXiv.org. doi: 10.48550/arXiv.math/0506090.
- [6] McLean, D J. (2020). *Animal trajectory analysis with trajr - trajr vignette*. Available at: <https://cran.r-project.org/web/packages/trajr/vignettes/trajr-vignette.html>.
- [7] Mandelbrot, M. (1967). How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension. *Science*, 156(3775), 636–638. doi: 10.1126/science.156.3775.636.
- [8] Sugihara, G, and May, R M. (1990). Applications of fractals in ecology. *Trends in Ecology and Evolution*, 5(3), 79–86. doi: 10.1016/0169-5347(90)90235-6.
- [9] Hartvigsen, G. (2021). *A Primer in Biological Data Analysis and Visualization Using R* (2nd ed). Columbia University Press. New York, NY. ISBN 9780231554404.
- [10] Murphy, K P. (2022). *Probabilistic Machine Learning: An Introduction* (2nd ed). MIT Press. Cambridge, MA. ISBN 978026236930.
- [11] Ceja, E G. (2021). *Behavior Analysis with Machine Learning and R: A Sensors and Data Driven Approach*. CRC Press. Boca Raton, FL. ISBN 978-1032067049. Available at: <https://enriquegit.github.io/behavior-free/>.
- [12] Kuhn, M. (2019). The caret package. Available at: <https://topepo.github.io/caret/>.
- [13] Biecek, P, and Burzykowski, T. (2021). *Explanatory Model Analysis: Explore, Explain, and Examine Predictive Models, With examples in R and Python*. CRC Press. Boca Raton, FL. ISBN 9780367135591.

Multi Repository Management Tools

Ulvi Shakikhanli ^{1*} and Vilmos Bilicki ¹

¹ Doctoral School of Computer Science, Faculty of Science and Informatics,
University of Szeged, H-6720, Szeged, Hungary

*E-mail of corresponding author: ulvi@inf.u-szeged.hu

Abstract: Multi repository is one of the most preferred structures in the development process. There are numerous tools available for managing Multi Repo projects. The use of these tools can affect the development period and productivity. In the first place, this paper explains Version Control Systems and the concept of branching together the most essential tools in this field. The second, the fingerprints or signatures of some of those tools are described for identifying repositories. Specific measurements have been calculated according to those tools and gave us a clearer view of the usage index of those tools. The study of actual data in the paper yields exact conclusions for the most essential features for Version Control Systems tools.

Keywords: Multi Repository Management, Version Control System, Github Mining, Development Productivity

Received 18 September 2022

Accepted 18 November 2022

Published 20 December 2022

1. Introduction

Multi Repository structures are popular among developer teams and companies. As a basic definition, it can be explained that the Multi Repository structure manages all packages and source code components in several repositories [1]. This allows each group of developers to focus on their part of the project and work independently. However, this approach introduces some difficulties. Since the project is separated into different modules, some tasks like testing, debugging, versioning, and even sometimes launching can create serious problems, which will delay the development. Comparison of the Multi Repository approach with Mono and other aspects of the Multi Repository structure was analyzed in detail in [2].

To overcome these and other related problems, software developers use Multi Repository Management Tools (MRMT). There is no unique description for this concept. The general definition is provided here for MRMT based on its main characteristics. MRMT is a software or service provider, which stores projects in one or more repositories and makes developer team collaboration possible. It is evident that MRMTs have much more capabilities than these two, but all of them are based on a basis of these two basic notions. Name storage of the project and help the developer team to work together.

Among the MRMTs, Github is the most popular one, and other tools may be split in two main categories: Major and Minor Multi Repository Management tools. Major tools have their own local server for storing repositories, and can work independently from Github. However, Minor tools have very limited feature capacity

and cannot work without Github. This paper focuses on the characteristics of these tools and present new findings about their general use. It is worth mentioning that there are dozens of such tools, but this paper will focus only on the most popular and promising ones. The followings are main questions which paper focuses on:

RQ 1: *What is version control?*

RQ 2: *What are the main features of Github?*

RQ 3: *Which VCSs have apparent advantages over Github?*

RQ 4: *Which tools increase Multi Repository Management process of Github, and what are their signatures?*

RQ 5: *What are the main features of Minor MRMTs?*

2. Literature review

Most of the information about Multi Repository management tools in the literature is present in different internet forums or websites rather than in academic papers. Most of the Version Control Systems (VCSs) may also be treated as MRMTs. In short, it can be said that VCSs permit the acceleration and simplification of the software development process, and allow new workflows [7]. There are two main approaches in VCSs namely, centralized and distributed. The centralized Version Control System (CVCS) stores the development history in a central server, while the Distributed one (DVCS) does the opposite; that is, it saves copies of repositories in all machines [8]. Each of them has its pros and cons. This is why the choice of what type of VCS should be used can affect the whole development period as regards the size and intensity of commits. Another article [9] widely explains at which level this can occur. However, the authors only briefly mention a few most popular tools, and this paper gives a more detailed review here. There are some of them, which had to be mentioned, for example, Github [10] and Mercurial [11]. These tools will be discussed in later sections in a more detailed way, but it has to be mentioned that Github has a



considerable role in the VCS field, with a market share of nearly 85% [12]. In this paper, Github Mining will be implemented to answer the previously mentioned questions. However, it is not an easy task. Since Github hosts millions of repositories, there are several cons about doing mining among them [13]. However, as described in the mentioned paper this problem can be avoided by implementing additional steps in presented algorithm.

3. Differences between mono and multi repository approaches

Starting from small individual projects to huge cooperative applications, the first task of developers is to choose the structure of the project. There are currently two main approaches to this. They are called Mono Repository, and Multi Repository approaches. Some big companies like Google have a long history of using the Mono Repository approach for their projects. Still the study described in [3] gives a clear view of the advantages and disadvantages of both methods. A comparison can be made according to the three main features of both methods, as provided in [3].

Visibility – According to the survey among Google developers, this may be considered the main advantage of having a Mono Repository structure. Developers mentioned the need for code to be accessible to all of them, which allows them to use the same code for all components and use the same development approach. Of course, this is also possible in the Multi Repository approach. Still, since each part of the project is located in a different repository, special permission has to be given.

Choice of developer tools, consistent style, and toolchain – As also mentioned in the article Google has created its own tools. Since the whole project is in a unique repository, this means that all the developers have to use the tools, which were created or recommended by Google. It is also regarded as one of the greatest tradeoffs of the Mono Repository approach. Developers mentioned in the survey that sometimes they were forced to use tools, that they were not familiar with or comfortable. It may be beneficial to use a unique style and development language for the whole project, but it restricts the freedom of the developers, and this can have adverse effects.

A version of dependencies – This is quite similar to above-mentioned themes. The Mono Repository approach forces developers to use the latest dependency and sometimes induces crashes or problems during the development phase. However, dependencies of different code portions do not vary in Multi Repository structures, and this allows developers to circumvent the diamond dependency problem.

Cognitive load issue - This issue is handled by both repository structures using different approaches. The Mono Repository approach attempts to achieve this goal by increasing the visibility of code and tools. At the same time, the Multi repo approach uses separate small codes, and this increases the build and development velocity.

Besides all these mentioned themes, both repository structure shows different behavior in several cases, which are described in [2]. In the end, it clearly can be said that

the choice between two-structure types is not a solid line and has to be decided by checking different tradeoffs.

4. Version control systems

During the development period, there may be situations where developers end up with thousands of code, and in some cases, they would like to check older versions of their source code. Version Control Systems (VCSs) are widely used for this purpose. It is worth to mention that VCS are tools, that keep the tracking information of files and they access and collaborate between developers. There are two main types of VCS, namely Centralized Version Control Systems and Distributed Version Control Systems.

Centralized Version Control Systems – uses one central repository approach.

With these, a central computer or server stores all the files and versions of the project. Users can access some specific files or entire repositories to work. After finishing their work, the user has to “*push*” their changes with a commit message. After pushing these changes, other users have to “*update*” their files according to the new version of the repository. It should be mentioned that CVCS stores just the last version of a project, so other users always have to keep up to date about changes in the source code. During the development phase, users can use branches to implementing new features or functionalities. Sometimes branches may crash, but it will not affect the work process of the whole project. *Branches* can be merged into the source code after the testing and implementation of features. *Branches will be discussed* in more detail below.

There are two main disadvantages of CVCSs, namely, single-point risk and low speed. As it mentioned above, CVCS contains only one central server for storing repositories. If it goes down, the whole project will be inaccessible, and the development process will stop. In addition, using one server creates another problem during the development phase. Users have to communicate with the central server for each command (create a branch, push, merge, and so on), and this creates a vast traffic overload for the server, and often induces a slow response from the server.

Distributed Version Control Systems - This is a new approach for CVCS, and does not require any central server. It means that every user can have a full copy of the repository and use it without asking for permission from the main server. It also runs faster than CVCS because almost all the commands run on the local machine, so users do not need any network connection. Most of the terms described in CVCS are also applied in DVCS. One of the most significant drawbacks of this approach may be memory consumption. Since it stores the whole project, it will consume a lot more memory, but DVCS uses compression for decrease the repository size. This approach is faster, more flexible, and reliable. If one repository goes down then one of the users can upload their own version within seconds. Because of all of these properties, DVCS is used a lot in the development industry nowadays, and there are massive platforms that provide these services. These include Github and Mercury.

5. Branching

A branch can be understood as a side development process independent from the main repository. Developers create a *branch*, an isolated workspace, from a particular state of the source code. They can share this branch and work on their tasks without affecting the rest of the project and later merge (or integrate) their changes back into the main line of development [4]. It is worth to mention that not all branches have to be merged to the main development line. In some cases, branches may be created just for testing purposes. Some of the different branching strategies will be explained in the below in more detail.

6. Branching strategies

The results of a survey are presented, based on interviews of developers and what they have learned about branching in the development process. From this, the basic idea about the purpose of branching and its types can be retrieved. After that, branching strategies will be explained in detail. As described in [5], there are several types of branches and they mostly have names related to their purpose. Five of the most common strategies:

1) *Release* - This branch type is mainly used for saving specific release versions of projects;

2) *Experiment* - It contains mostly experimental codes and implementations which are not included in the release version;

3) *Feature* - It is for implementing new features. It had to be merged only after the testing and verification phases;

4) *Bug fix* - As suggested by the name, this branch is created for fixing bugs and errors in projects;

5) *Contributor* - This branch is mainly used by individual developers to separate their work before sharing it with others.

Besides all of these branching types, several clear branching strategies are used to improve productivity and the quality of collaboration during the development phase.

The article the blog [6] describes the main characteristics of the branching strategy:

- It provides a clear path for the development process from initial changes to production;
- It allows users to create workflows that lead to structured releases;
- It permits parallel development;
- It optimizes developer workflow without adding any overhead;
- It enables faster release cycles;
- It efficiently integrates with all DevOps practices and tools, such as different versions control systems.

Three main branching strategies:

Git Flow - this strategy supports multi branches for managing source code. There are two main groups for these branches:

a) Primary branches - They are divided into two parts: master, which keeps the main production code, and

development, which is created for implementing the development process.

b) Support branches - They are divided into three parts: feature branches for implementing new features, a hotfix is for dealing with errors and bugs, and release, which is for preparation of release.

The advantages of this strategy are standard support among Git tools, a clear naming strategy, and being ideal while handling multiple version development. On the other hand, disadvantages can be that history becomes a bit complex and hard to read and not very useful when developers need one version of release.

Github Flow - As suggested by the name, the strategy is offered by Github itself. With this strategy, there is one main master branch, and it is used as the main development branch. Users have to create additional branches for bug and feature implementation and all of them at the end have to be merged to the main branch after successful testing. The advantage of this strategy are that it is clear and straightforward. The git history will not become more complex, as it has been mentioned above. However, simplicity may also be a disadvantage for this strategy because it is unsuitable for a difficult development process.

It should be added that besides these popular features, there might be several other strategies based on the development culture of the company and the tool used for version control.

7. Major multi repository management tools

As described in the literature review, Github is one of the most popular VCSs in the market.

7.1. Main features of Github

Several main features make it so popular. These are:

a) Collaborative coding - Since it is one of the most essential features of VCS, Github focuses on this area. It enables substantial developer teams to work together. There are several UI elements and tools for tracking the work of other developers, and this allows the whole team to work in a synchronized way. Due to this feature, developers can add comments to their and changes of other developers, which increases teamwork and development quality.

b) Automation - this feature mainly focuses on the automation of testing, building, labelling and issue labeling. Developers can add commands for testing, and other phases of automation, which saves enormous amounts of time and increases the quality of code besides keeping the build process in line.

c) Security - This feature combines several aspects of the previous two. It provides critical reviews and code scanning. Although Github hosts a vast amount of open source, projects it also has a Private repository feature which is mainly designed for companies and commercial projects.

d) Project management - As it is understandable from its name, this feature helps developers manage their repositories. It should not be mixed up with MRMT, which was presented earlier. It covers managing a single repository, its commits, and labels, and so on.

Besides these features, Github has a few more features, it provides all these almost free, and this has attracted millions of developers and created a vast developer community. Since it is so popular among developers, it stores more than 128 million public repositories. Because of this, even other big MRMTs store their repositories in Github.

VCSs like Beanstalk, Helix core, Mercurial and Gerrit have some advantages over Github. All of these tools use Github as storage, but they also have public and private servers, which allows them to store their repositories. These features are listed below:

Beanstalk - Unlike Github, it mainly focuses on private repositories, and this is why Beanstalk gets business more than Github.

Helix core - One key difference is that it is centralized and not distributed like Github. It means Helix core stores all project files in one machine so not all developers will have a copy of the whole project as they do in Github.

Gerrit - The code review is a key advantage of Gerrit over Github. Unlike Github, where users can have only one commit in a pull request, so the developer will not get confused when they come to check it.

Mercurial - The learning period for Mercurial is much simpler than Github. Mercurial is viewed by many as more secure than Github. However, it should be mentioned that the same security level can also be achieved in Github, but it takes more effort and requires more knowledge.

7.2. Main aspects of MRMTs

During analysis of the features of the above-mentioned tools, it became evident that some of them have similar aims, and this could prove helpful in identifying their main aspects. In addition, these aspects may be crucial because they highlight the weaknesses of Github.

1) *It updates several repositories at the same time* - This is essential for the majority of tools. They do it in different ways, but in the end, all of them help users to update, (by “update” it means pull, push, etc.) several repositories at once. It saves a lot of time for developers and project owners, which overall increases the development period. It can also cause some problems in

big projects, but most of the tools have different approaches to reduce this theatre.

2) *It monitors the development process* - Github gives the user a detailed view of the development period and collaboration of developers among themselves. Still it cannot do the same with several repositories at the same time. Many tools were created for this purpose. They give a clear view of the contributors, work description of individual developers, commits, and so on. This information becomes crucial during the management of Multi Repository projects. The quality of this information may be different for each tool, but this is undoubtedly valuable since it is not possible to get it in Github.

3) *It manages third-party elements*. This part is not fully covered in the managing Multi Repository project, but it is helpful in this area as well. Managing dependencies and different packages may become a nightmare owing to the MR project development, and there is no proper command or tool inside the Github platform to handle it. However, the tools, which are mentioned here, can do it quickly, and they can save a lot of time for developers.

4) *It improves review and pull requests*. This feature is not common among MRMTs, but there are several tools, that do a great job and highlight the weakness of Github and other services. It is mainly used for big projects with tens of contributors. In this case, it becomes impracticable for project owners to manage all the pull requests, and perhaps not every contributor updates their projects according to new merges. Hence, this functionality seeks to overcome this issue by making the process more straightforward and more understandable.

5) *It manages the project build process*. This is quite similar to the automation feature of Github. The main difference here is that the tools focus on building several repositories at the same time to make projects compatible.

In Figure 1, it can be seen how popular these features are among users. The number of users has been calculated for each tool. It should be added that some tools may use one or more features from above list. If so, the user share of this tool will be added to all the features to maintain fairness. The features will be represented by their ordinal number given in the previous section.

Table 1. Github Multi Repository Management procedures.

Name	Feature	Signature
Mepo	Update, compare, compere, save	components.yaml
MR	Update, list, offline	.mrconfig
Pull	Update (all forks)	pull.yml
West	Combine all repos, update, list	west.yml
Mention-bot	Improve review and pull requests	.mention-bot
Zappr	Improve review and pull requests	.zappr.yaml
Mrgit	Manage project build process	mrkit.json
Talan CLI	Manages third party elements	.tln.conf
DevOps & Swarm-mode	Manage project build process	docker-compose.swarm.yml
Lerna	Monitor development process	lerna.json

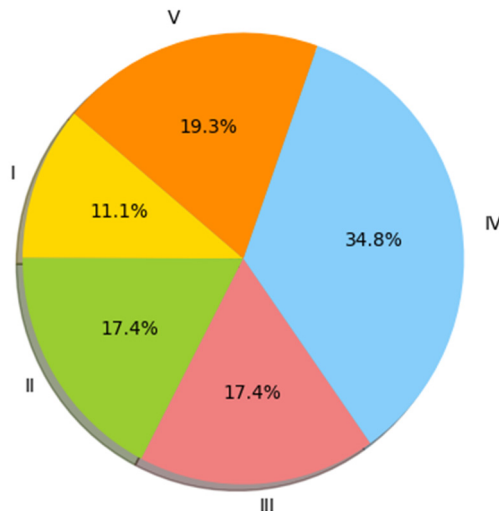


Fig. 1. Percentage of features.

Figure 1 can provide a good overview of which features are needed most for MRMTs. Nearly 35% of the users employ tools with the *Improve review and pull request feature*. This may be the greatest weakness of the Github platform since there is a considerable demand for tools like these. There are several reasons for this but the main one is the branching system. Of course, it is a key feature, but this brings several problems. For example, there may be vast amounts of branches in Multi Repository projects, and the review and pull request process can become a challenge for managers in this scenario. The percentage of the other three features is almost the same. This means that features like the *Monitoring the development process, managing third-party elements, and managing the project build process* have nearly the same importance for users, but none of them is as important as the fourth feature. The least important feature is the first one presented in the last section. It is surprising that updating all the repositories at once is not a priority for users.

These results may be helpful for a developer who wants to create a new MRMT. It offers us a clear view of the needs of the market and users, and it highlights the weakness of popular VCS and MRMTs.

8. Measurements of MRMTs

The database was created using the signatures of the tools above-mentioned and different statistics can be obtained with the help.

1) Usage of tools over the years.

Figure 2 provides a precise picture of the usage level. As can be seen, there was a considerable increase in both 2017 and 2022.

This paper analyzed projects only created until October of 2022 but still usage of tools in this year is significantly higher than others are. One of the main reasons for that can be increase in the usage of VCSs and different MRMTs. In addition, it seems that different weaknesses of several Major MRMTs are become more obvious, and developers rely more on 3rd party Minor tools. First years during the appearances of these tools,

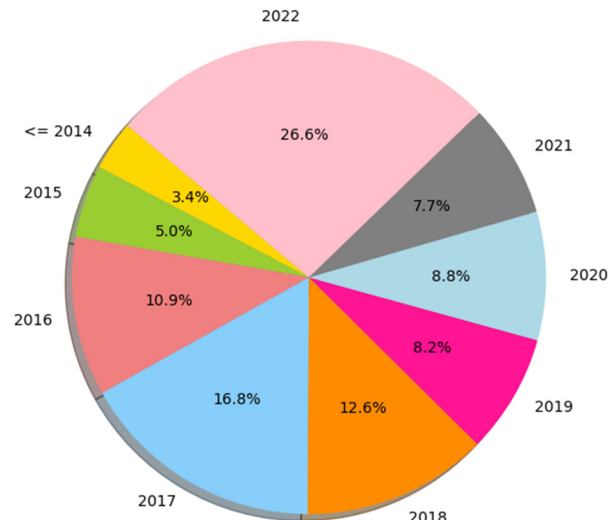


Fig. 2. The general usage percentage of tools.

most of them have been developed by individual developer by non-commercial purposes, and that can be reason for low usage of these tools until 2016. Between the years 2016-2018 usage of tools increased significantly comparing previous years and that is the period where several problems with Major VCSs become more and more obvious. Starting from 2019 the usage of these tools decreased again, this is the same period when Github added new features, which more or less solved some problems, and Minor tools again lost their popularity.

All of these changes during the years shows the importance of Minor tools in this field. Some tools for example "Pull" even inspired Github to add similar features and one can say that these tools has a huge impact shaping this industry.

9. Discussion and conclusion

Here, the VCSs, MRMTs, branching, and different branching strategies have been reviewed. It should be admitted that most of these have already been presented in other research studies. Still none explained concepts like branching and versioning together and revealed connections between them as it done here. There are some excellent VCSs and MRM (Multi Repository Management) tools like Github, Mercury and Beanstalk. Still here, this paper also presented some less-known yet effective tools for Multi Repository management. They can boost productivity of development with a big community and several powerful features, but they have five main drawbacks, which are also the main features of minor MRMTs. These are the following:

i) It should update several repositories at once - Which is not possible in major VCSs, and such features can only be incorporated by implementing minor MRMTs like "Mepo", "MR", and "Pull".

ii) When it comes to monitoring the development process - Github has some UI elements needed to support this feature. Still it lacks cohesion between different repositories at once, and developers need to use other tools like "Lerna".

iii) It should manage third-party elements - Unfortunately, major MRMTs, including Github, do not pay too much attention to this topic, and leave it to the

discretion of project owners. This is why some less important tools like “Talan CLI” or “Lerna” have been presented to solve this issue.

iv) Improve review and pull request – Here, the importance of pull request and branching in the first section have been mentioned and primary tools have some disadvantages in this area too. Pull requests may become a challenge when there are tens of branches, and merge requests and this often creates bottlenecks and reduces the speed of development. Tools like “Mention-bot” and “Zappr” which are presented before could overcome this problem.

v) Manage project build process - This function usually causes problems during the build process, as suggested by its name. Tools like Github use *Automation* to solve it, but it is not effective when there are several repositories, and all of them have to be used at the same time. “DevOps & Swarm-mode” and “Mrgit” are the tools that may be useful in such cases.

Besides introducing less essential tools and their properties, this paper also presented “*signatures*” of these tools and provided sound methods for other researchers interested in this area. In the last section, two fundamental values were introduced, and they tell us something about the importance of tools and their use ratio over the past few years.

This paper has analyzed several terms like *Version Control*, *Branching*, *Branching strategies* and *Minor Multi Repository Management tools*. Main features of VCSs have been analyzed. A unique Database has been created and several analyses, which has never been shown or discussed before, have been introduced in this paper. It has been proved that Minor tools has huge impact on VCSs, and therefore has to be taken account during project development process.

Threads to validity

It has to be taken account that measurements shown in this paper are created according to the collected repositories from Github. As it has been described in previous chapters, those repositories are identified according to the *signature files* presented in repository folder. However, it is known that some tools do not necessary need some configuration tool. Even in some cases, there is no such file, and that is why identifying usage of those tools inside repository is almost not possible. In addition, this is a rapidly growing field of programming and new tools emerge almost each month. Thus, popularity ratio of tools and needs of MRMT industry can be much different in upcoming years.

References

- [1] Scott, P L. (2017). *Mono-repo or multi-repo? Why choose one, when you can have both?* Medium.com, 2017. [Online]. Available at: <https://patrickleet.medium.com/mono-repo-or-multi-repo-why-choose-one-whe>.
- [2] Shakikhanli, U., and Bilicki, V. (2022). Comparison between mono and multi repository structures, *Pollack Periodica*, vol. 17, issue 3, pp. 7-12. doi: <https://doi.org/10.1556/606.2022.00526>.
- [3] Otte, S. (2009). *Version Control Systems*, Available at: https://www.mi.fu-berlin.de/inf/groups/ag-tech/teaching/2008-09_WS/S_19565_Proseminar_Technische_Informatik/otte09version.pdf.
- [4] Rama Rao N. and Chandra Sekharaiah K. (2016). A Methodological Review Based Version Control System with Evolutionary Research for Software Processes. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16)*. pp. 1–6. <https://doi.org/10.1145/2905055.2905072>.
- [5] Brindescu, C. C., Codoban, M., Shmarkatiuk, S. and Dig, D. (2014). How do centralized and distributed version control systems impact software changes? In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. pp. 322–333. <https://doi.org/10.1145/2568225.2568322>.
- [6] Github. <http://www.github.com/> Accessed March, 2022.
- [7] Mercurial. <https://www.mercurial-scm.org>. Accessed March 25, 2022.
- [8] Slintel. <https://www.slintel.com/tech/source-code-management/github-market-share>. Accessed March 20, 2022.
- [9] Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., and Damian, D. (2014). The promises and perils of mining GitHub, *MSR 2014: Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 92–101, <https://doi.org/10.1145/2597073.2597074>.
- [10] Jaspan, C., Jorde, M., Knight, A., Sadowski, C., Smith, E. K., Winter, C., and Murphy-Hill, E. (2018). Advantages and disadvantages of a monolithic repository: a case study at google. *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '18)*, pp. 225–234. <https://doi.org/10.1145/3183519.3183550>.
- [11] Barr, E.T., Bird, C., Rigby, P.C., Hindle, A., German, D.M., Devanbu, P. (2012). Cohesive and Isolated Development with Branches. In: de Lara, J., Zisman, A. (eds) *Fundamental Approaches to Software Engineering. FASE 2012. Lecture Notes in Computer Science*, vol. 7212. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-28872-2_22.
- [12] Phillips, S., Sillito, J. and Walker, R. (2011). Branching and merging: an investigation into current version control practices. *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, pp.9–15. <https://doi.org/10.1145/1984642.1984645>.
- [13] *DevOps Branching Strategies Explained*: Available at: <https://www.bmc.com/blogs/devops-branching-strategies/>, Accessed March 22, 2022.

Solar Panel Tracker System Using GPS Technology

Alina Fazylova ^{1*}, Anuar Kussainov ¹, Akzhaina Baidullina ² and Asel Smailova ²

¹ Institute of Telecommunications and Space Engineering,
Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeyev,
Almaty, Kazakhstan

² Institute of Electrical Engineering and Electrotechnics,
Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeyev,
Almaty, Kazakhstan

*E-mail of corresponding author: sharf_06@inbox.ru

Abstract: This article provides a solution to the problem of not improving three-dimensional systems with light sensors in tracker systems for solar panels, due to the inefficiency of operation in cloudy weather, with an increase in precipitation and contamination of photodetectors. The paper shows the calculation of the position angle of the tracker system relative to the changing position of the sun. The GPS system is able to automatically determine the location of the solar panel (latitude, longitude), then the microcontroller with the developed program performs a mathematical calculation, and the system automatically adjusts the solar panel to the sun. Thus, the need to use a photo sensor is eliminated, since the system automatically determines the optimal angle between the sun and the solar panel.

Received 18 November 2022

Accepted 29 December 2022

Published 1 February 2023

Keywords: Tracker, Solar panel, Photodetector, Microprocessor, Real-time module.

1. Introduction

Today there is a problem in the efficiency of solar panels. The use of solar trackers increases the efficiency of solar panels by 30% due to the possibility of full tracking of the Sun. But for the most part, such trackers work thanks to the use of various sensors tracking the Sun. Such sensors are very susceptible to climatic conditions, they also have a high level of wear, work differently in different regions and are not always accurate [1].

The demand for solar energy is constantly growing, since solar energy is environmentally friendly and affordable. The second reason for the relevance of the use of solar energy is its resource intensity. In just 9 minutes, the Earth receives more energy from the Sun than humanity produces in an entire year. This energy is supplied free of charge and has no impact on the environment.

Solar energy is an area of significant investment in conditions of declining oil and gas reserves. Solar energy contributes to the reduction of global consumption of fossil fuels.

Solar panels for the production of solar energy can be installed on your own roof. The generated energy can be converted into heat and stored in a battery that will be used for heating and water supply.

The solar power plant contains a solar panel with a biaxial orientation system to the sun, on which photovoltaic modules containing linear photodetectors are

installed as sensors for tracking the sun. Signals from photodetectors with the help of a microprocessor control the drives of the system of vertical and horizontal orientation of solar panels [2].

The disadvantage of this installation is the low accuracy of tracking the sun, and the photo sensors are also exposed to external influences. They get dirty, dusty, which leads to inaccuracy of their work. With a small cloud cover, the photo sensors cannot catch sunlight and do not turn the panel in the sun. This leads to losses of generated energy. When the photo sensor is set to high sensitivity, the tracker can turn the solar panel to moonlight, which leads to a waste of energy on turning the solar panel [3].

2. Materials and methods

The basis of the tracker device for GPS-based solar panels is the ARDUINO UNO microcontroller, since this controller meets the requirements for speed and the number of output pins. The real-time clock module c or its analog DS3107 is also used. Both modules work fine without any changes in the sketch. The system uses GPS in order to synchronize the real-time module to correct the course of the DS3107 once a month, these time inaccuracies do not have a large deviation value and, accordingly, do not affect the operation of the device as a whole, since there is no need for high accuracy of time measurement. The HX1230 display was used to display the information. The software provides for the input of the tracker response period, depending on the area of use, from 5 to 55 minutes. With the onset of sunset, the tracker automatically turns to sunrise, and is in standby mode. In this version of the software, you can use both a single-axis mechanism and a two-axis one. Before



© The Author(s). Published by CIEES Association Press. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 Generic License.

developing the entire solar panel tracker system and modeling the overall scheme, it is necessary to connect and test each module of the control system separately. The GPS receiver allows you to determine the location of the object using GPS (global positioning system), this receiver must be connected to the Arduino board. GPS (Global Positioning System) (Figure 1).

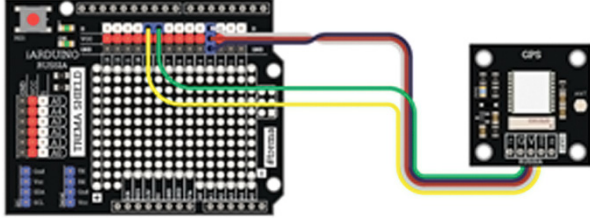


Fig. 1. GPS connection diagram.

The system also uses a real-time module (Figure 2). The following requirements are imposed on the real-time module: the calculation of the current time and date must be carried out continuously and regardless of the shutdown of the system.

The system also uses a real-time module (Figure 2). The following requirements are imposed on the real-time module: the calculation of the current time and date must be carried out continuously and regardless of the shutdown of the system.

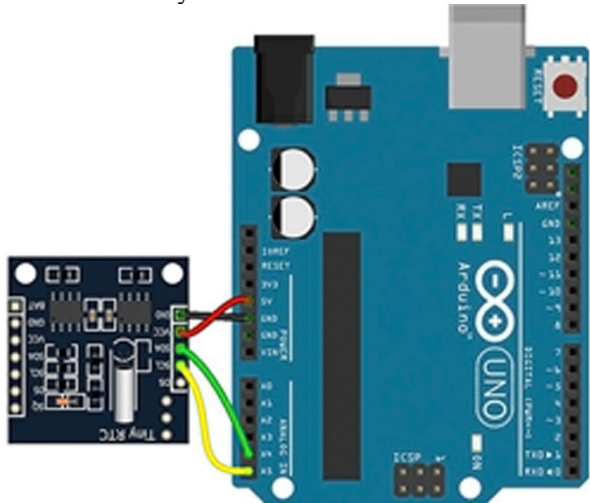


Fig. 2. Connection diagram of the real-time module.

The DS1307 module is one of the most affordable and common RTC modules. It can accurately track seconds, minutes, hours, days, months and years. Some of the important features of the DS1307 are:

- The ability to generate a programmable rectangular signal

- Low current usage; less than 500 nA in battery backup mode;

- Ability to set the date to 2100;

- I2C serial interface.

The HX1230 LCD screen is a basic graphic LCD screen. Its connection diagram is shown in Figure 3.

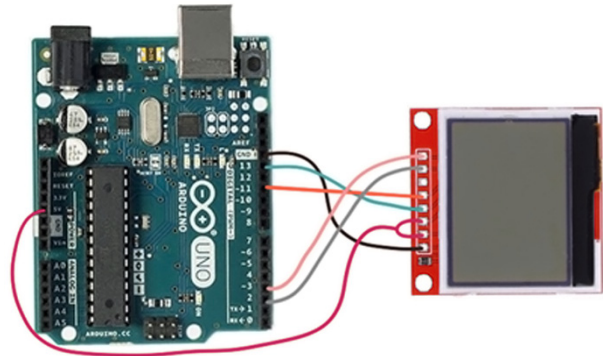


Fig. 3. Electrical diagram of the HX1230 LCD screen.

2.1. Calculation of the maximum and minimum azimuth of the Sun

The calculation of the maximum and minimum azimuth of the Sun at a given location is determined using a number of initial and calculated data, which include: Latitude (Ξ); Longitude; (Φ) Time zone (ω); Date (d); Time (τ); Specified Julian day (ν); Calculated Julian century (δ); Mean geometric longitude of the Sun related to the mean equinox (p); Mean anomaly of the Sun at time (ξ); Eccentricity of the Earth's orbit at time (μ); Calculated equation of the center of the Sun at time (λ); True longitude (k); True anomaly (l); Radius-vector of the Sun (heliocentric distance from the Sun to the Earth from center to center) (θ); Correction for nutation and aberration (n); Inclination of the ecliptic (ζ); Tilt coordinates (ϵ); Mean right ascension (asc); Sunset (dec); Dispersion (var); The angle of the solar azimuth (azimuth).

The mean geometric longitude of the Sun refers to a certain period of time, which is called an epoch. We are currently using the epoch J2000.0. This epoch began on January 1, 2000 at noon Coordinated Universal Time. At that moment, the average longitude of the Sun was: $280^{\circ} 27' 59.26''$ (or 280.46646 degrees). The mean geometric longitude of the Sun is to find where the average value of the Sun was at time T , referring to the date of the equinox.

$$\rho = (280.46646 + \delta(36000.76 + 0.00030\delta)) \quad (1)$$

The average anomaly of the Sun at a time. The word "anomaly" here refers to the uneven or "anomalous" apparent motion of the Sun and planets along the plane of the ecliptic. The mean anomaly is defined as "the angular distance from the perihelion that [the Earth] would have if it were moving around the Sun at a constant angular velocity." Further, using the data obtained from the first formula, it will be possible to calculate the true longitude and anomaly by relating it to the equation of the center of the Sun.

$$\varepsilon = 357.52911 + \delta(35999.05029 - .0001537\delta) \quad (2)$$

The eccentricity of the Earth's orbit at a time. Eccentricity refers to the "flatness" of an ellipse outlined by the Earth in its orbit around the Sun. The initial constant 0.01670834 used is almost unchanged due to tiny corrections in the rest of the calculation equation. As

an example: if the time T corresponds to the launch of the satellite in 1957, then $e = 0.01672$. The Earth's orbit tends to be closer to a circle than to a very flat ellipse. That's why it's a small value.

$$\mu = 0.016708634 - \delta \left(\frac{0.00042037 +}{+0.00000012\delta} \right) \quad (3)$$

The calculated equation of the center of the Sun at a time. The equation of the center in this solution will be a different version of the Kepler equation. This condition is suitable when the eccentricity of the Earth's orbit is very small. (As calculated above, the Earth's orbit is the case when $0 < e < 1$ and $e = 0.01672$).

$$\lambda = \sin(\epsilon^{rad})(1.914602 - \delta(0.004817 + 0.000014\delta) + \sin(2\epsilon^{rad})(0.019993 - 0.000101\delta) + 0.000289 \sin(3\epsilon^{rad})) \quad (4)$$

True longitude

$$k = \rho + \lambda \quad (5)$$

The true anomaly

$$l = \epsilon + \lambda \quad (6)$$

The radius vector of the Sun is the distance (measured in astronomical units or AU) between the center of the Sun and the center of the Earth (the calculations use a value slightly less than one, namely 0.997).

$$\theta = \frac{1.000001018(1-\mu^2)}{1+\mu\cos(l^{rad})} \quad (7)$$

Making adjustments for nutation and aberration. This item provides greater accuracy when detecting solar coordinates using true longitude. In fact, it is necessary to calculate a value that corrects the true longitude of the Sun taking into account perturbations (such as the tidal force of the Moon on Earth).

$$\eta = k - 0.0056 - 0.0047\sin((125.04 - 1934.1\delta)^{rad}) \quad (8)$$

The slope of the ecliptic describes the slope of the ecliptic. Tilt is an effect caused by the tilt of the Earth around its axis relative to the celestial equator. In other words, it is the angle between the plane of the earth's equator and the plane on which the Sun and planets seem to move.

$$\zeta = 23 + \frac{26 + \frac{21.448 - \delta(46.815 + \delta(0.00059 - \delta(0.001813))}{60}}{60} \quad (9)$$

The tilt coordinates will be detected by the Sun's the coordinates of the inclination of the Sun, as a right ascension and declination of the apparent position of the Sun on the celestial sphere at a time, were found through parallax correction and the introduction of these corrected values into the two main calculation functions.

$$\epsilon = \zeta + 0.00256 \cos((125.04 - 1934.136\delta)^{rad}) \quad (10)$$

Solar Right Ascension (asc)

Right ascension is one of the two polar coordinates in the rotating equatorial coordinate system of spherical astronomy. Its second coordinate is declination. Right

ascension in this case will be the angle measured at the celestial equator between the point of the vernal equinox and the intersection point of the hour circle passing through the measured celestial body. This is an analogue of the longitude of a place on the globe (second coordinate: latitude). Right ascension is more often expressed in time than in degrees or radians. The specification in terms of time refers to the fact that the apparent rotation of stars around the Earth is proportional to time and that the time difference can be determined more easily than the angles.

$$\text{asc} = \arctg\left(\frac{\cos \eta^{rad}}{\cos \epsilon^{rad} * \sin \eta^{rad}}\right) \quad (11)$$

Sunset (dec)

In this case, sunset implies an angle based on the coordinates of the inclination of the Sun, taking into account the correction for nutation and aberration.

$$\text{dec} = \arcsin(\sin \epsilon^{rad} * \sin \eta^{rad}) \quad (12)$$

Variance (var)

In physics, dispersion is usually understood as the deviation of an object due to interaction with another local object (scattering center). Examples are the scattering of light on atoms or fine dust, or neutrons on atomic nuclei. The strength of the dispersion is determined by the so-called dispersion cross section. The name comes from the fact that the scattering cross section according to the classical theory, the dispersion of material points on a solid sphere is just equal to the cross section of the sphere. Elastic and inelastic dispersion are distinguished. In the case of elastic dispersion, the amount of kinetic energy after the collision remains unchanged. In the case of sunlight, the dispersion is correlated to inelastic. In the inelastic dispersion, part of the available kinetic energy is converted into the excitation energy of the atom or used to break the bond, for example, in ionization processes. The variance calculation provides information about the shape of the interaction potential.

$$\text{var} = \text{tg}\left(\frac{\epsilon^{rad}}{2}\right) * \text{tg}\left(\frac{\epsilon^{rad}}{2}\right) \quad (13)$$

Equation of Time (eq)

The equation of time in this calculation refers to the angular ratio of the mean geometric longitude and the mean geometric anomaly of the Sun in the corresponding coordinates and, consequently, the definition of the eccentric orbit of the Earth:

$$\text{eq} = 4(\text{var} * \sin(2 * \rho^{rad}) - 2 * \mu * \sin(\epsilon^{rad}) + 4 * \mu * \text{var} * \sin(\epsilon^{rad}) * \cos(2 * \rho^{rad}) - 0.5 * \text{var}^2 * \sin(4 * \rho^{rad}) - 1.25 * \mu^2 * \sin(2 * \epsilon^{rad})) \quad (14)$$

Sunrise Equation (ha)

The calculation of the optimal location of the Sun at the desired point at a given time, these parameters are calculated by correlating the coordinates of the location of the object and the results of the "Sunset" equation.

$$\text{ha} = \arccos\left(\frac{\cos(90.833^{rad})}{\cos(\epsilon^{rad}) * \cos(\text{dec}^{rad})}\right) - \text{tg}(\epsilon^{rad}) * \text{tg}(\text{dec}^{rad}) \quad (15)$$

Local sunny noon (noon)

$$\text{noon} = \frac{740 - 4 * \phi - eq + \omega * 60}{1440} \quad (16)$$

Local sunrise time (rise)

$$\text{rise} = \frac{\text{noon} - ha * 4}{1440} \quad (17)$$

Local sunset time (set)

$$\text{set} = \frac{\text{noon} + ha * 4}{1440} \quad (18)$$

Duration of sunlight (dur)

The duration of sunlight or the duration of effective insolation is a climatic indicator that measures the length of time during which a place is exposed to effective insolation, for example, solar radiation is intense and strong enough to create significant shadows. This indicator can be used as a reference for estimating the frequency of "good weather".

The duration of sunlight varies from region to region depending on geographical factors such as latitude, longitude and altitude, and climatic factors such as cloud cover or precipitation. The duration of the sunshine more or less strictly follows the general geographic distribution with a zonal trend, i.e. by latitude, but the small-scale distribution remains much more complex.

$$\text{dur} = 8 * ha \quad (19)$$

True Solar Time (tru)

The equation of true solar time based on the coordinates of the location of the object and the equation of time according can be calculates as follow:

$$n^I = \frac{0.1}{24} \quad (20)$$

$$\text{tru} = \frac{n^I + \frac{0.1}{24} * 1440 + eq + 4 * \phi - 60 * \omega}{1440} \quad (21)$$

Hour angle (ang)

The hour angle, together with the declination and distance (from the center of mass of the planet) determines the position of the celestial object, is the angle of a large circle passing through the object and the two celestial poles. Thus, it is a higher concept that takes into account the terrain and the depth to the center of the Earth at the location of the ground observer. In particular, the hour angles are the angles of ideal circles perpendicular (at right angles) to the celestial equator. The declination of an object observed on a celestial sphere is the angle of this object to/from the celestial equator (i.e. in the range from +90 ° to -90 °).

Condition:

$$\frac{\text{tru}}{4} < 0 \quad (22)$$

$$\text{true: ang} = \frac{\text{tru}}{4} + 180 \quad (23)$$

$$\text{false: ang} = \frac{\text{tru}}{4} - 180 \quad (24)$$

The angle of the solar zenith (zen)

The angle of the solar zenith is the angle between the sun's rays and the vertical. It is closely related to the angle of elevation of the sun, that is, the angle between

the sun's rays and the horizontal plane. Since these two angles are additional, the cosine of one of them is equal to the sine of the other. Both of them can be calculated using the same formula using the results of spherical trigonometry. On a sunny afternoon, the zenith angle is minimal and equal to latitude minus the angle of declination of the sun.

$$\text{zen} = \arccos(\sin(\mathcal{E}^{\text{rad}}) * \sin(\text{dec}^{\text{rad}}) + \cos(\mathcal{E}^{\text{rad}}) * \cos(\text{dec}^{\text{rad}}) * \cos(\text{ang}^{\text{rad}})) \quad (25)$$

The angle of elevation of the Sun (elev)

The angle of elevation of the Sun above the horizon of the observer is called the height of the sun; in particular, the maximum height of the sun at noon (12 hours, true solar time), the measurement of which is the easiest way to determine the latitude of the location. The height of the sun at any time of the day can be calculated from an astronomical triangle using the time, date and position of the observer.

$$\text{elev} = 90 - \text{zen} \quad (26)$$

Approximate atmospheric refraction (approx)

Astronomical refraction is a change in the refraction of a beam of light falling on the earth from the outside, due to refraction in a stratified atmosphere. The reason is an increase in the refractive index from $n=1$ in the vacuum of space to about $n=1.00029$ on earth. Astronomical refraction is a special case of terrestrial refraction. The curvature of the light rays is directed downward — in the same sense as the curvature of the Earth, but much less. The most severe curvature occurs near the earth and makes up a maximum of 10-15% of the curvature of the earth with a very flat sight.

The Earth's atmosphere is denser at the surface of the earth than at high altitudes. Therefore, light rays coming from space at an angle are increasingly deflected downward due to astronomical refraction. To an Earth observer, if they are not approximately at the zenith, the stars appear higher than they would be without the Earth's atmosphere. The refraction value depends on the tangent of the zenith distance, as well as on the temperature and air pressure at the observer's location. At an altitude of 5 km, about half of the value at sea level is reached. The reason for astronomical refraction is the refraction experienced by each ray of light during the transition from an optically thinner medium to a denser one. The change in the direction of propagation occurs in differentially small steps between adjacent layers of air (Snellius' law of refraction) and must be integrated along the entire light path.

Condition:

$$\text{elev} > 85$$

$$\text{approx} = 0 \quad (27)$$

$$5 < \text{elev} < 85$$

$$\text{approx} = \frac{58.1}{\text{tg}(\text{elev}^{\text{rad}})} - \frac{0.07}{(\text{tg}(\text{elev}^{\text{rad}}))^3} + \frac{0.000086}{(\text{tg}(\text{elev}^{\text{rad}}))^5} \quad (28)$$

$$-0.575 < \text{elev} < 5$$

$$\text{approx} = \frac{1735 + \text{elev} * (-518.2 + \text{elev}(103.4 + \text{elev}(-12.79 + \text{elev} * 0.7)))}{3600} \quad (29)$$

$$\text{elev} < -0.575$$

$$\text{approx} = \frac{\frac{-20.772}{\text{tg}(\text{elev}^{\text{rad}})}}{3600} \quad (30)$$

Solar altitude adjusted for atmospheric refraction (refr)

$$\text{refr} = \text{elev} + \text{approx} \quad (31)$$

The angle of the solar azimuth (azimuth)

The angle of the solar azimuth is the azimuth angle of the position of the Sun. This horizontal coordinate determines the relative direction of the Sun along the local horizon, and the zenith angle of the sun (or an additional angle of elevation of the sun) determines the apparent height of the Sun.

The following formulas assume clockwise rotation. The angle of the solar azimuth can be calculated with a good approximation using the following formula.

The formulas below can also be used to approximate the azimuth angle of the Sun, but these formulas use cosines, so the azimuth angle displayed by the calculator will always be positive and should be interpreted as an angle between zero and 180 degrees, in cases where the hour angle is negative (morning), and the angle between 180 and 360 degrees, when the hour angle is positive (during the day) (These two formulas are equivalent if we take the approximation formula "angle of elevation of the Sun").

Condition:

$$\text{ang} > 0$$

$$\text{azimuth} = \frac{\arccos\left(\frac{\sin(\xi^{\text{rad}}) * \cos(\text{zen}^{\text{rad}}) - \cos(\text{dec}^{\text{rad}})}{\cos(\xi^{\text{rad}}) * \sin(\text{zen}^{\text{rad}})}\right)}{360} \quad (32)$$

$$\text{ang} > 0$$

$$\text{azimuth} = \frac{540 - \arccos\left(\frac{\sin(\xi^{\text{rad}}) * \cos(\text{zen}^{\text{rad}}) - \cos(\text{dec}^{\text{rad}})}{\cos(\xi^{\text{rad}}) * \sin(\text{zen}^{\text{rad}})}\right)}{360} \quad (33)$$

3. Research results

It is necessary to draw up dependency relationship diagrams based on the previously given calculation points

The diagram of the position of the Sun is represented by the ratio of the solar height adjusted for atmospheric refraction and the angle of the solar azimuth (Figure 4).

The diagram of the position of the sun is valid only for locations with the same geographical latitude and only for one location if it is parameterized not by local solar time (true or average), but by zonal time. The position of the sun at any time of the day and on any day of the year, as well as the change in the duration of the sunshine can be read on the diagram of the position of the sun for this place.

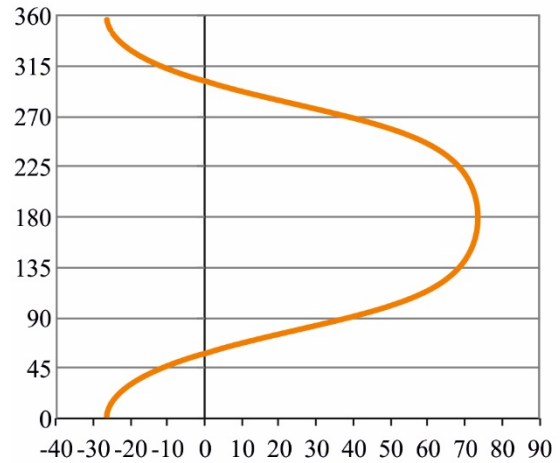


Fig. 4. Position diagram.

The solar declination diagram (Figure 5) indicates the declination of the sun, i.e. the time and latitude at which the sun is at its zenith, adjusted for nutation and aberration.

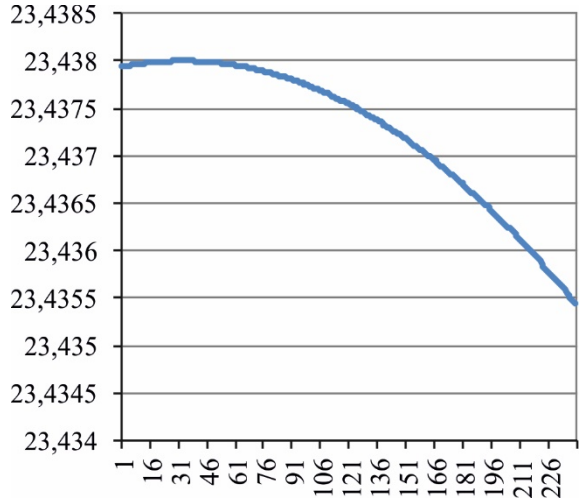


Fig. 5. Solar declination (deg.).

The diagram of the ratio of solar altitude adjusted for atmospheric refraction and time after midnight at a given location (Figure 6). This diagram is necessary to determine the point of maximum elevation of the Sun and the virtualization of the equipment using this calculation.

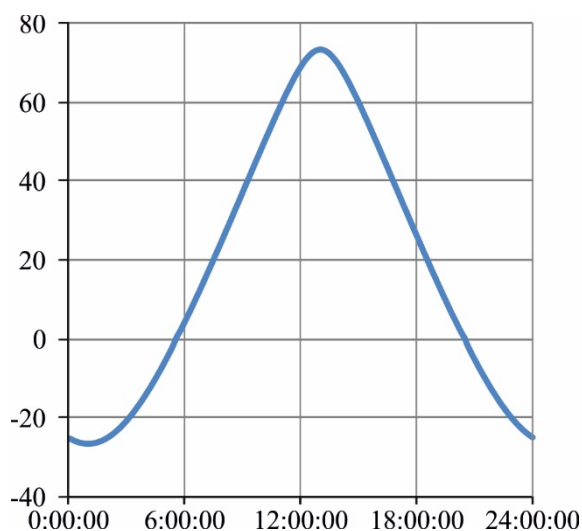


Fig. 6. The ratio of height to time.

4. Discussion and conclusion

In this manuscript a new approach to the development of a solar panel guidance system on the sun was developed. According to the proposed approach a method for calculating the guidance of a solar battery on the sun has been elaborated. The angle of the tracker system position relative to the changing position of the sun is calculated.

According to the proposed mathematical equations have been calculated the maximum and minimum azimuth of the sun, and the following parameters:

Solar right Ascension; Sunset; Dispersion; Equation of time; Equation of sunrise; Local solar noon; Local

sunrise time; Local sunset time; Duration of sunlight; True solar time; Hour angle; Angle of solar zenith; Angle of elevation of the Sun; Approximate atmospheric refraction; Solar altitude adjusted for atmospheric refraction.

References

- [1] Gliberman A. Ya. and Zaitseva A. K. (1961). *Silicon solar panels*. Moscow: Gosenergoizdat, 74 p. (in Russian)
- [2] Pavlov N. (2013). Solar energy - the energy of the future, *Electronics: science, technology, business*. vol. 1 issue 123, pp. 130-137
- [3] Umarov G. Ya., and Ershov A. A. (1974). *Solar energy*. M.: Znanie, 64 p. (in Russian)
- [4] Kharchenko N. V. (1991). *Individual solar installations*. M.: Energoatomizdat, 208 p. (in Russian)
- [5] Kashkarov A. P. (2011). *Wind generators, solar panels and other useful structures*. M.: DMK Press, 144 p. (in Russian)
- [6] J. F. Lee and N. A. Rahim, (2013). "Performance comparison of dual-axis solar tracker vs static solar system in Malaysia," *2013 IEEE Conference on Clean Energy and Technology (CEAT)*, Langkawi, Malaysia, pp. 102-107, doi: 10.1109/CEAT.2013.6775608.
- [7] GOST R 57229-2016. *Photovoltaic systems. Sun tracking devices. Technical conditions*. – M.: Standard inform, 64 p.3.
- [8] Ya'u M. J. (2017). A Review on Solar Tracking Systems and Their Classifications, *Journal of Energy, Environmental & Chemical Engineering*. vol. 2, issue 3, September 2017, pp. 46-50. doi: 10.11648/j.jeece.20170203.12
- [9] Stadnik V.V. and Razgonyaev Yu.V (2008). Assessment of the total arrival of solar radiation entering inclined surfaces, *Proceedings of the Main Geophysical Observatory named after A.I. Voeikov*. issue 557, pp. 67-84.
- [10] <http://www.macsun solar.com> / Accessed March 30, 2022.

Statistical Analysis of Counter-Covid-19 Policies Adequacy and Effectiveness in Croatia

Filip Šklebar¹, Marko Mikša¹ and Robert Spudic^{1*}

¹ Krapina University of Applied Sciences/Laboratory for Spatial Intelligence, Krapina, Croatia

*E-mail of corresponding author: robert.spudic@vhz.hr

Abstract: Countries around the world have imposed various counter-measures at national level, adjusting societal and economic activities to constrain the Covid-19 pandemic. Here we report the research results of assessment of adequacy and effectiveness of the Covid-19 pandemic counter-measures in the case study of the Republic of Croatia. Statistical analysis and inference methods are applied to identify the adequate national-level Covid-19 counter-measures, and their effective intensity levels, as a lessons-learned evidence with potential use for policy and strategy development support, in potential similar future situations.

Keywords: Covid-19, pandemic, counter-measures, adequacy, effectiveness.

Received 18 March 2023

Accepted 27 April 2023

Published 15 May 2023

1. Introduction

The Covid-19 pandemic has exposed the world to unprecedented challenges of managing socio-economic processes at national and international levels. Countries have deployed various counter measures, adjusting societal and economic activities to constrain the pandemic.

Advances in information and communication sciences have allowed for collection, aggregation, and analysis of massive data sets of public health and societal activity indicators. Those include the structured observations of counter measures set on national levels to mitigate the pandemic.

Here we report the research aimed at assessment of adequacy and effectiveness of the Covid-19 pandemic counter measures in the case study of the Republic of Croatia. Adequacy is understood as the ability to enforce the targeted counter measure with the appropriate or productive intensity. Effectiveness is considered the ability of the counter-measure studied, and imposed at the adequate level of intensity, to counter the daily number of newly infected individuals at the national level. We also address of feasibility of utilization of a single socio-economic indicator, encompassing a number of specific ones, in association/correlation with the increase or decrease in number of new daily infections, nation-wide.

The presented research aims at identification of productive counter measures, and their productive intensity levels, as a lessons-learned experience with potential use for policy- strategy-development support, in potential similar future situations.

The manuscript reads as follows. Section 2 outlines studied counter measures and statistical methods used to assess their adequacy and effectiveness. Section 3 presents research results in the forms of figures and tables

with appropriate explanations. Section 4 discusses research results, outlines the conclusions drawn, and summarizes contributions of the presented research to the general understanding of the subject.

2. Method and material

Croatia has a population of 3.899 million. It is situated in south-eastern Europe, while extending a diverse topography, which determines Croatia's socio-economic framework, as depicted in Figure 1.

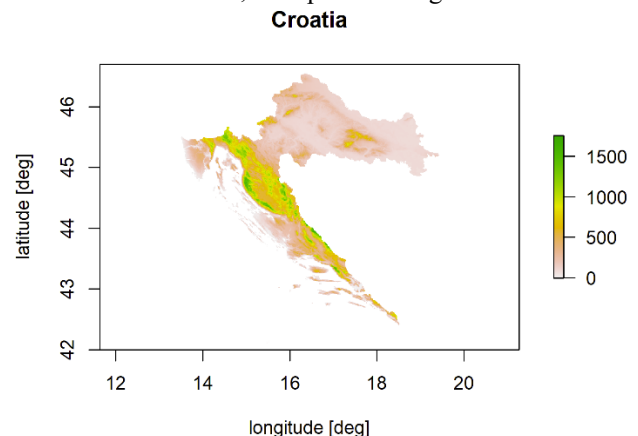


Fig. 1. Map of Croatia.

The economic profile of Croatia comprises activities yielding profit in tourism (15.9 % national GDP in 2021.[1]), agriculture and food industry.

Majority of Croatian citizens travel for the purpose of business, education, recreation, and health. Patterns of migrations range from daily migrations for work and business, weekend migrations for education (students traveling home from major cities where they study) and weekend recreations, and seasonal ones (tourism).

The Covid-19 counter measures in Croatia tackles physical gatherings of individuals with the specific/unique effects on social and economic activities nation-wide.



The first cases of COVID-19 in Croatia were reported in late February 2020. By late-January 2022, cumulative infections exceeded 899174 with about 13566 deaths. The previous peak of new daily cases was reached in mid-November 2021, with about 5527 new cases reported daily. Total number of reported cases was 1.27 million and deaths 18127.[2]

We utilize the daily number of new infections as collected by Our World in Data [2], and associate it with the database of counter measures and their intensity levels, as reported in database Coronavirus Government Response Tracker managed by University of Oxford. [3].

We identify adequacy and effect of the imposed counter measures, with the aim to identify those producing the containment results.

This Section outlines the methodology, and describes the original data used in the presented research.

2.1 Material

Data sets examined in the presented research comprises those collected from Websites Oxford Coronavirus Government Response Tracker [3] and from Our World in Data [2], with methodology and data description. Time series examined in the paper: (1) Daily number of new Covid-19 cases in Croatia (nc), (2)

Economic Support, Index (ESI) levels of provision of government funding assistance to economy and society, (3) Cancellation of Public Events (CPE) level imposed, (4) School Closing (SC) level, (5) Stringency Index – Average (SI), as a collated indicator of the imposed policies, (6) Restrictions on Gathering (RoG) level, (7) Closing Public Transport (CPT) level, and (8) International Travel Controls (ITC) level. Economic Support Index (ESI) is calculated on a scale of 0 to 100, with higher scores indicating more comprehensive and effective economic support provided by governments support to its citizens in response to the COVID-19 pandemic [4]. The ESI is a composite index that considers four main categories of economic support measures: income support, debt relief, fiscal measures, and monetary measures [4]. The Cancellation of Public Events (CPE) index is a measure developed by the Blavatnik School of Government at the University of Oxford to track the extent to which governments around the world have canceled or modified public events in response to the COVID-19 pandemic [4]. The CPE index is based on data collected from publicly available sources, including government websites, news reports, and social media [4].

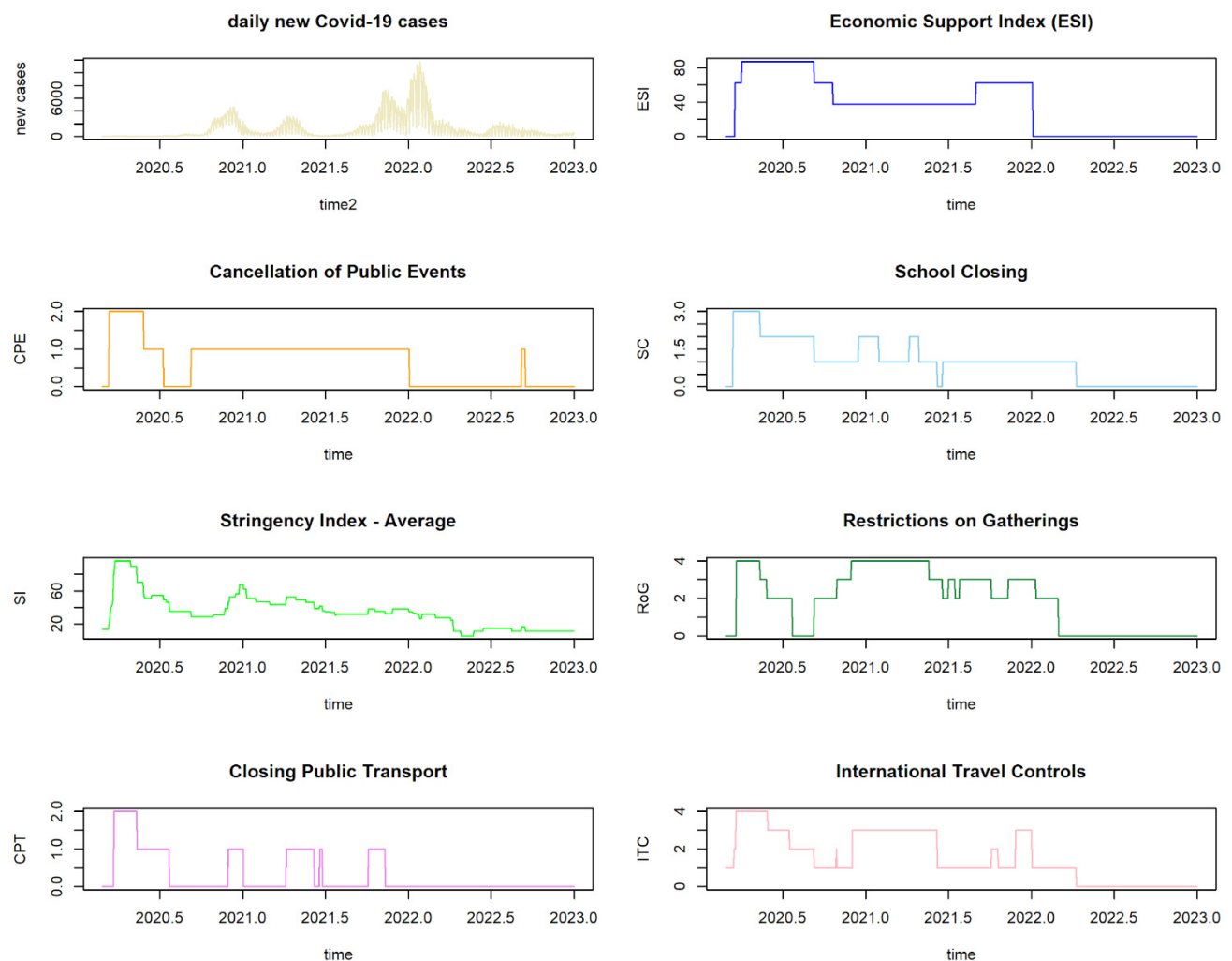


Fig. 2. Time series of indices considered.

The index is calculated on a scale of 0 to 100, with higher scores indicating more extensive cancellation or modification of public events [4]. The School Closing (SC) level is a system used to indicate the severity of measures implemented to close or partially close schools during a pandemic or other emergency situation [4]. The SC level is typically based on a scale from 0 to 4, with each level indicating progressively more severe measures from no closures or restrictions in place, schools are open and operating as usual (SC level 0) to the total closures or restrictions in place where schools are closed entirely and all learning takes place online (SC level 4) [4].

The Stringency Index (SI), developed by the Blavatnik School of Government, measures the strictness of government policies in response to the COVID-19 pandemic [4]. It uses a scale from 0 to 100, with higher scores indicating more stringent measures [4]. It is based on a composite measure of government policies that restrict people's behavior during the COVID-19 pandemic and combines nine response indicators, including school closures, workplace closures, travel bans, public events, and gathering restrictions, among others [4]. The Restrictions on Gathering (RoG) level is a system used to indicate the severity of measures implemented to restrict social gatherings during a pandemic or other emergency situation [4]. The RoG level is typically based on a scale from 0 to 4, with each level indicating progressively more severe restrictions from level 0 with no restrictions on gatherings in place to the level 4 with all gatherings prohibited [4]. Closing Public Transport (CPT) refers to the decision to suspend or stop public transportation services in a particular area, city, or country due to pandemic and security concerns [4]. The typical CPT levels are 0 - No measures, 1 - Recommend closing (or significantly reduce volume/route/means of transport

available) and 2 - Require closing (or prohibit most citizens from using it) [4]. International Travel Controls (ITC) measures include travel bans and quarantine requirements [4]. The ITC dataset is made from information on the implementation date, duration, and scope of the measures [4]. These measures are recorded on a scale from 0 to 4, where 0 indicates no restrictions and 4 indicates a complete ban on all international travel [4].

Time series of indices considered are shown together in the chart Figure 2.

A combined view of several results of exploratory statistical analysis is presented in Figure 3. The scatter-plot diagram is found in the lower triangle, which shows the fit between pairs of variables. Diagrams on diagonal depict estimates the experimental probability density function curve for every variable considered. The upper triangle contains correlograms of pairs of indices considered. None of the considered indices follows normal statistical distribution.

Statistical significance of the determined Pearson correlation coefficients is expressed in graphical terms based on the related statistical test p-values, as follows: (i) *** denotes $0 < p < 0.001$, (ii) ** denotes $0.001 < p < 0.01$, (iii) * denotes $0.01 < p < 0.05$, (iv). denotes $0.05 < p < 0.1$ and (v) no mark denotes $0.1 < p < 1$.

The analysis established that policy/measure indices are mostly categorical or discrete, except for SI, which can be expressed as a continuous variable. Overall, there is weak correlation between the target (nc) and the predictors (policy/measure indices), and modifications to policy/measure indices do not appear to have a substantial effect on the target variable.

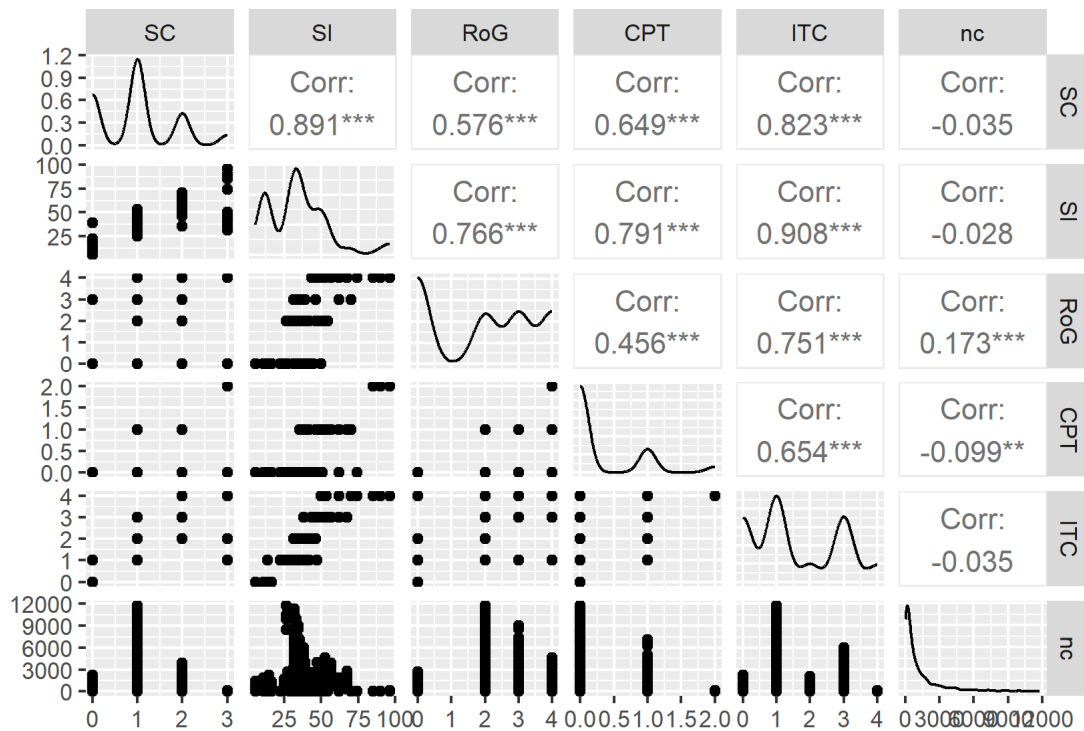


Fig. 3. Combined scatter-plot diagram, and correlogram of indices considered.

The high correlation between policy/measure indices suggests that they are interconnected and have a linear relationship. As a result, modifications to one policy/measure index could potentially affect the other indices. A normal distribution is commonly used to represent normal distributions, however, in this research, all the indices under consideration follow a non-Gaussian distribution, implying that the data may be skewed. The SI index summarizes the effects of individual policy/measure indices and it is hypothesized that there is a correlation between the target variable (nc) and the SI index. Therefore, changes to the SI index may have a more substantial impact on the target variable (nc) than changes to individual policy/measure indices.

2.2 Method

Time series of daily number of new Covid-19 cases in Croatia, and intensities of the eight counter-measures, listed in Section 2.1. are examined in the presented research.

Data are assessed using the common methods of exploratory statistical analysis, including the correlation analysis [5]. Inference on adequacy and effectiveness of counter-measures are drawn using examination of diagrams of pairs of time series, scaled to reveal the meaning. Time series of daily numbers of new Covid-19 cases is matched with the scaled time series of counter-measure intensities, separately.

Association between daily number of new Covid-19 cases in Croatia, and the Stringency Index, presumed a single indicator of the overall mitigation intensity by a set of countermeasures, is examined using correlation analysis [5], as well as using lagged Granger Causality Index [6]. Finally, statistical significance of every counter-measure intensity in description/prediction of daily number of new Covid-19 cases is examined by formal application of linear regression model development method [7].

The variable importance of the predictor is examined using Linear regression model (LRM) importance of predictors. [8] Variable importance is calculated as an absolute t-value obtained from Formal linear regression model.

3. Research results.

During the Covid-19 outburst in Croatia daily number of new cases was low. Still, Croatian government decided to impose very strict measures to counter the spread of the virus. This decision was made with the rationale of the unknown characterization of the virus and means of its spread.

Later phases of Covid-19 pandemic were not tackled with such a one-side approach, but in consideration of epidemiology, public health system, and socio-economic effects.

The presented research assesses adequacy and effectiveness of particular counter measures on the daily number of new Covid-19 cases through analysis of scaled time series diagrams of such pairs.

Scaled time series of Economic Support Index (ESI, blue line) and the daily number of new Covid-19 cases are depicted in Figure 4.

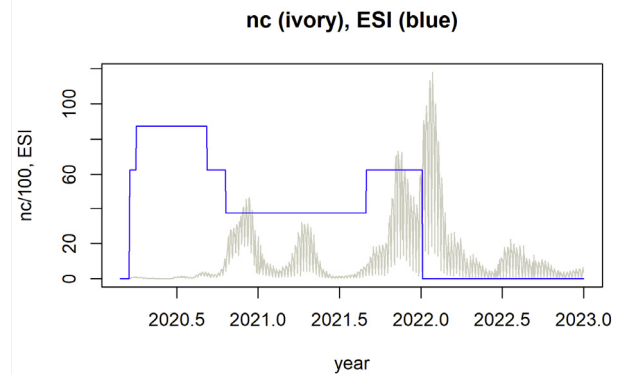


Fig. 4. Effects of economic support index.

Initially, the government decided to strongly support the economy in order to preserve the jobs. There were a significant number of businesses which involved close contact with larger numbers of people such as bars and restaurants that were closed. One of the aspects of this measure was also aimed to ease the acceptance of the lockdown. As the number of new cases continued to be low it was decided to open some of the businesses which enabled the government to lower the support. This resulted in an increased number of new cases. When the number of new cases increased rapidly the government decided to increase the support which was not as efficient as in the beginning of the pandemic when it mitigated the spread of the virus.

Scaled time series of the effect of school closure (SC, skyblue line) and the daily number of new Covid-19 cases are depicted in Figure 5. School closure at sufficient level (Level 2 and above), mitigates the transmission of disease. Lectures were mainly organized online with the exception of the exams. Since online lectures reduced the contact between lecturers and students, the quality of the process suffered. The government was closely monitoring the number of new cases and lowered the level of school closing with a decreased number of new cases. It can be seen that insufficient level of School closure may even instigate disease.

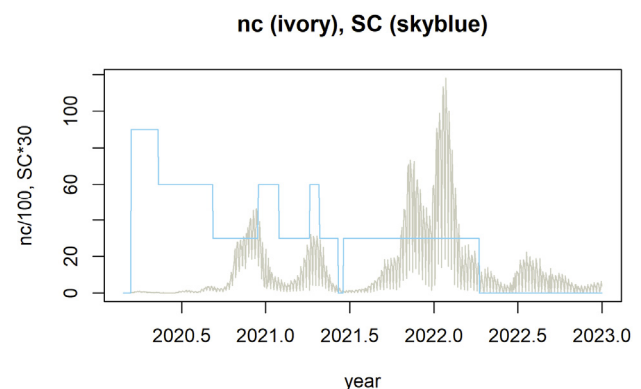


Fig. 5. Effects of school closure.

Time series of the effect of the Stringency Index (SI, green line) and the daily number of new Covid-19 cases are depicted in Figure 6. Stringency Index higher or equal to 60 tackles the outburst efficiently and successfully. Lower SI levels may be efficient on smaller outbursts.

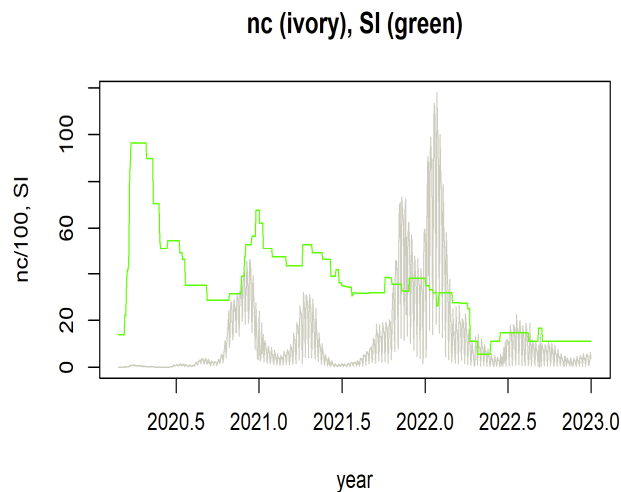


Fig. 6. Effects of stringency index – average.

Scaled time series of the effect of Reduction of Gatherings (RoG, spring green line) and the daily number of new Covid-19 cases are depicted in Figure 7. Level 3 and above may be seen as successful in mitigation of outbreaks. Lowering levels in situations of raising the number of infections produces counter effects, instigating the outbreak.

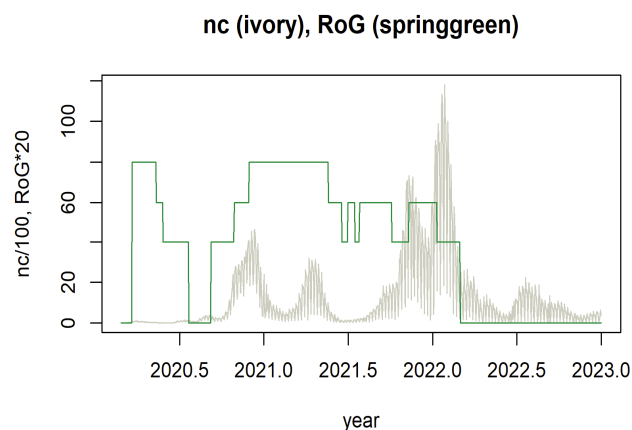


Fig. 7. Effects of reduction of gatherings.

Scaled time series of the effect of Closing Public Transport (CPT, violet line) and the daily number of new Covid-19 cases are depicted in Figure 8. Even a moderate level of public transport suspension appears as efficient in tackling the outbreak, despite the mandatory face masks usage.

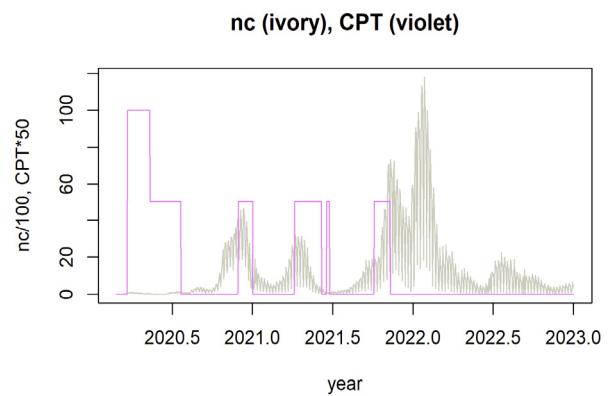


Fig. 8. Effects of closing public transport.

Scaled time series of the effect of International Traffic Controls (ITC, red line) and the daily number of new Covid-19 cases are depicted in Figure 9. A sufficient level (Level 3, or above) should be imposed to tackle the outbreak efficiently. ITC of a sufficient level should be imposed for a month, at least, to produce effect.

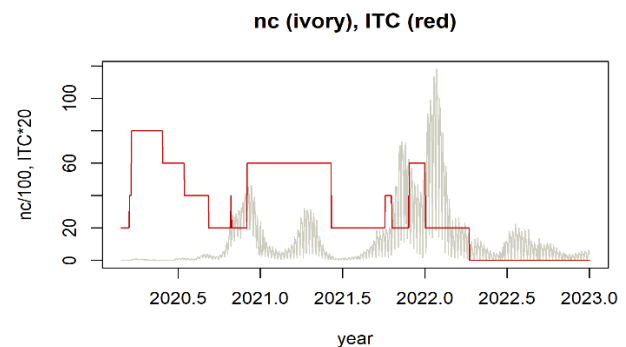


Fig. 9. Effects of international traffic controls.

The cause-effect relation between numeric indices of new cases and Stringency Index is examined using lagged Granger Causality Index, as depicted in Figure 10. No formal effects caused on the number of new cases by the Stringency Index was found.

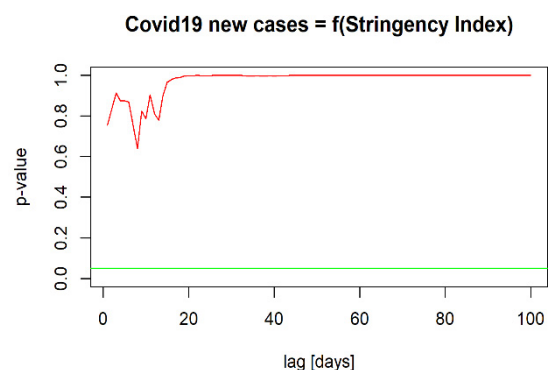


Fig. 10. Cause-effect relation between numeric indices nc and SI.

The cause-effect relation between numeric indices of Stringency Index and new cases is examined using lagged Granger Causality Index, as depicted in Figure 11. Formal effects of the number of new cases on Stringency Index found with a delay of approximately 3 months.

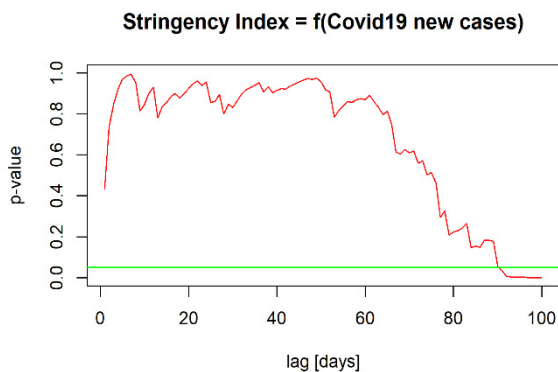


Fig. 11. Cause-effect relation between numeric indices nc and SI.

Analysis of the linear regression model of nc as a function of SI showed poor performance. SI is not a statistically significant predictor of nc. The Breusch-Godfrey test with a lag order of 10 was used to detect the presence of auto correlation in the residuals of a regression model. [9] Test result returned p-value less than $2.2e-16$. Low p-value indicates that there is evidence of serial correlation in the residuals.

Furthermore, various structures of the linear regression models emphasize particular indicators over SI in description/prediction of daily number of new Covid-19 cases, and, consequently, the adequacy and efficiency of the applied counter-measures.

Formal Linear Regression Model with all predictors concerned was introduced. Results are depicted on Figure 12. Model gives a relatively good bias description. Model also gives poor variance description. Larger p-values suggest that CPT is not statistically significant and SI is barely significant.

```
Call:
lm(formula = nc ~ ., data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-2654.0  -850.9  -386.8   211.0   8790.5

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1377.122    170.916   8.057 2.14e-15 ***
ESI          -11.147     2.793  -3.991 7.05e-05 ***
CPE         -1428.211    166.072  -8.600 < 2e-16 ***
SC           936.887    182.337   5.138 3.31e-07 ***
SI           -30.982    12.968  -2.389 0.017064 *
RoG          956.153    68.093  14.042 < 2e-16 ***
CPT          276.319    181.227   1.525 0.127637
ITC          -388.745    102.940  -3.776 0.000168 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1659 on 1033 degrees of freedom
Multiple R-squared:  0.2102, Adjusted R-squared:  0.2048
F-statistic: 39.27 on 7 and 1033 DF, p-value: < 2.2e-16
```

Fig. 12. Formal Linear Regression Model with all predictors concerned.

Similar analysis was done with the CPT variable removed from the model, as depicted in Figure 13. This model gives a relatively good bias description. Model also gives poor variance description. Larger p-values suggest that SI is not statistically significant.

```
Call:
lm(formula = nc ~ ESI + CPE + SC + SI + RoG + ITC, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-2685.9  -852.9  -401.9   253.0   8832.2

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1203.652    127.629   9.431 < 2e-16 ***
ESI          -10.652     2.776  -3.837 0.000132 ***
CPE         -1427.274    166.177  -8.589 < 2e-16 ***
SC           830.441    168.550   4.927 9.72e-07 ***
SI           -17.534     9.512  -1.843 0.065566 .
RoG          910.022    61.041  14.908 < 2e-16 ***
ITC          -412.490    101.820  -4.051 5.48e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1660 on 1034 degrees of freedom
Multiple R-squared:  0.2084, Adjusted R-squared:  0.2038
F-statistic: 45.37 on 6 and 1034 DF, p-value: < 2.2e-16
```

Fig. 13. Formal Linear Regression Model with all predictors concerned, except CPT.

Similar analysis was done with the CPT and SI variable removed from the model, as depicted in Figure 14. This model gives a relatively good bias description. Model also gives degraded variance description ability. All predictors are statistically significant.

```
Call:
lm(formula = nc ~ ESI + CPE + SC + RoG + ITC, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-2802.3  -865.8  -430.6   267.7   8895.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1033.323    88.141  11.724 < 2e-16 ***
ESI          -9.357     2.689  -3.480 0.000522 ***
CPE         -1553.417    151.608  -10.246 < 2e-16 ***
SC           622.321    125.292   4.967 7.95e-07 ***
RoG          883.977    59.452  14.869 < 2e-16 ***
ITC          -507.517    87.907  -5.773 1.03e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1662 on 1035 degrees of freedom
Multiple R-squared:  0.2058, Adjusted R-squared:  0.202
F-statistic: 53.64 on 5 and 1035 DF, p-value: < 2.2e-16
```

Fig. 14. Formal Linear Regression Model with all predictors concerned, but CPT, SI.

Linear regression model (LRM) importance of predictors is depicted in Figure 15.

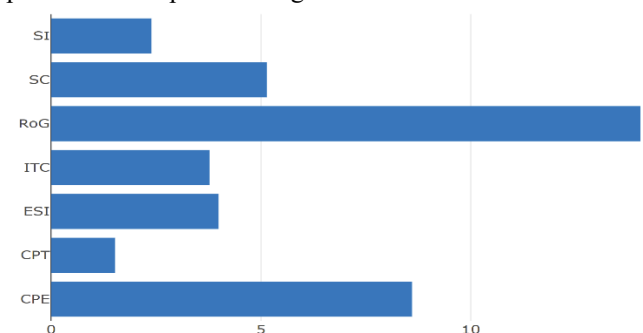


Fig. 15. LRM importance of predictors.

Importance of predictors is calculated as an absolute t-value from Formal Linear Regression Model with all predictors concerned.

Reduction of Gatherings and Cancellation of Public Events are the most efficient policies in tackling outbreaks. Imposing the others may have the expected

and wanted effects, when applied in particular timing, and the enforcement level.

4. Discussion and conclusion.

The presented research aimed at identification of productive counter measures, and their productive intensity levels, as a lessons-learned experience with potential use for policy- strategy-development support, in potential similar future situations.

The presented research aimed at assessment of adequacy and effectiveness of the Covid-19 pandemic counter measures in the case study of the Republic of Croatia. Scaled time series of the counter measures were compared with a number of new cases to assess adequacy of each countermeasure. Statistical significance of effectiveness of a counter-measures was assessed with Formal Linear Regression model. Reduction of Gatherings and Cancellation of Public Events are the most efficient policies in tackling Covid-19 outbreaks.

References

- [1] Ministry of Tourism and Sports of the Republic of Croatia, (2022). *Turizam u brojkama 2021*. Available at: https://mint.gov.hr/UserDocsImages/2022_dokumenti/Turizam%20u%20brojkama%202021.pdf, Accessed on April 23, 2023.
- [2] Mathieu E., Ritchie H., Rodés-Guirao L., Appel C., Giattino C., Hasell J., Macdonald B., Dattani S., Beltekian D., Ortiz-Ospina E. and Roser M. (2020). *Coronavirus Pandemic (COVID-19)*. Available at: <https://ourworldindata.org/coronavirus>, Accessed on April 23, 2023.
- [3] OxCGRT project. (2023). *Oxford Coronavirus Government Response Tracker (OxCGRT) data*. Available at: <https://github.com/OxCGRT>, Accessed on February 12, 2023.
- [4] Hale, T. et al. (2022). *Variation in government responses to COVID-19 BSG-WP-2020/032 Version 14.1*. Blavatnik School of Government, University of Oxford. Oxford, UK. Available at: <http://www.bsg.ox.ac.uk/covidtracker>, Accessed on February 12, 2023.
- [5] Mills, T. C. (2019). *Applied Time Series Analysis: A Practical Guide to Modeling and Forecasting*. Academic Press. London, UK.
- [6] Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3), 424–438. doi:10.2307/1912791.
- [7] Montgomery D. C., Peck E. A. and Vining G. G. (2012). *An Introduction to Linear Regression Analysis*. John Wiley & Sons, Inc.
- [8] Thomas, D.R., Hughes, E. and Zumbo, B.D. (1998). On Variable Importance in Linear Regression. *Social Indicators Research* 45, pp. 253–275, <https://doi.org/10.1023/A:1006954016433>
- [9] Breusch, T.S. and Godfrey, L.G. (1978) Misspecification Tests and Their Uses in Econometrics. *Journal of Statistical Planning and Inference*, 49, pp. 241-260, [https://doi.org/10.1016/0378-3758\(95\)00039-9](https://doi.org/10.1016/0378-3758(95)00039-9).

Towards an Approach to Improve an Environmental Management System in Industrial Organization

Neli Babekova ^{1*}

¹ Department of Management and Business Development,
University of Ruse “Angel Kanchev”, Ruse, Bulgaria

*E-mail of corresponding author: nbabekova@uni-ruse.bg

Abstract: The article presents an approach to improve an implemented environmental management system in industrial organization based on the self-assessment using a methodology in terms of specifying the level of maturity through such model of environmental management systems, based on ISO 14001 as a factor for development of a sustainable development of organizations from the mineral raw materials industry in Bulgaria. A practical application of the methodology in "Kaolin" EAD - a leader in the "Industrial Minerals" sector is presented. The described methodology ensures an opportunity to the interested parties of the company which operates in an environmentally friendly way that annually have to assess the current level of maturity and proposes actions that will lead to the sustainable development of its activity

Keywords: Maturity model, Environmental management systems, Methodology, Mineral raw materials industry.

Received 29 March 2023

Accepted 26 April 2023

Published 15 May 2023

1. Introduction

An industry is a group of enterprises, which operate in a particular business field. Manufacturing industry is divided into two major groups– extractive and producing [1, 2].

The extractive industry operates with extraction of fuel, minerals and other natural materials from the Earth's subsurface, while the producing industry consists of enterprises engaged in their processing.

Part of the extractive industry is the mineral industry. It covers the extraction of ore and industrial minerals.

The mineral industry of the Republic of Bulgaria is a structurally established industry and is one of the main drivers of economic development through increasing the consumption of raw materials from the domestic market and the production of finalized products with high added value [3]. The mining industry develops through efficient, complex and long-term utilization of underground resources in accordance with the requirements for sustainable development, covering the three main aspects: economic, environmental - green mining industry, corporate social responsibility.

This industry has a direct impact on nature - it changes entire landscapes, ecosystems and habitats. At the same time, it has great potential - to reclaim land, to positively impact the ecological status of large areas and populations of valuable species. It ranks among the most significant polluters of water and soil, and generates hazardous and recyclable waste and rock. Soil and water quality protection and proper waste management are a top

priority for achieving sustainable development.

The goal of the article is to outline an approach to environmental management system improvement, applying self-assessment using an environmental management system maturity model as a factor for achieving sustainable development of organizations in the mineral industry in Bulgaria. An application of the Methodology for this self-assessment in an organization from the Industrial Minerals sector is presented.

2. Towards an environmental management system

Objective of any organization is to achieve sustainable development in a complex and dynamic environment. In order to achieve it securely, senior management strives to meet the needs and expectations of all stakeholders. In addition, the drive is aimed at systematically improving the overall performance of the organization, which includes to plan, implement, analyse, evaluate and improve an efficient and effective management system [4].

The Environmental Management System (EMS) covers the organisational structure, planning activities, responsibilities, procedures, processes and resources for the development, implementation, enforcement, evaluation of the environmental policy [5]. The implementation of an EMS according to ISO 14001 is voluntary and demonstrates an organisation's concern for reducing pollution by controlling the impacts it has on the environment through its activities, products and services. ISO 14001 helps to develop a framework for an organisation's overall approach to the environment by requiring the company to define its environmental policy and objectives and the procedures that support them.



Implementing and certifying an Environmental Management System allows organizations to manage their environmental issues, leading to better environmental performance, as well as minimizing the risk of environmental incidents and their consequences [6].

The overall environmental performance of mining industry organisations is better managed through the development and implementation of an Environmental Management System in accordance with ISO 14001 [7].

Part of the minerals industry is the Industrial Minerals Sector. The leader in this sector is the company Kaolin EAD. It has implemented and operates an environmental management system in accordance with BDS EN ISO 14001:2015.

The development, implementation and operation of the Environmental Management System through the application of a systematic approach aims to achieve the following objectives:

- to establish an appropriate environmental policy;
- to ensure, relevant to the companies' production activities and products and environmental aspects, scope and content of the Environmental Management System
- to achieve and maintain compliance with all applicable national and local environmental requirements;
- to achieve good environmental performance through supervised activities, products or services, taking into account environmental objectives and policy;
- to prioritise and set reasonable, achievable and economically viable environmental goals and targets as an element of the organisation's Business Plan and investment programmes;
- to continuously improve environmental management practice (EMP) by carrying out

systematic inspections and implementing corrective and preventive actions, and consider changing circumstances.

The elements of the Environmental Management System of Kaolin EAD are shown in Figure 1. It demonstrates the approach on which the environmental management system is based - Plan-Do-Check-Act (PDCA) concept. The PDCA model provides an iterative process that is used by the organisation for continuous improvement. It includes:

- Planning: defines environmental objectives and the processes required to achieve them in accordance with the organisation's environmental policy;
- Implementation: implementation of processes as planned;
- Verification: monitoring and measuring processes against environmental policy and targets and reporting results;
- Action: take action for continuous improvement.

The figure represents premises of BDS EN ISO 14001:2015 and the respective steps of the PDCA model.

Step one, 'planning' identifies the organization's context, environmental policy and objectives, environmental aspects, compliance obligations, and action planning to achieve environmental objectives.

Step two 'implementation' includes sustainment and activities such as resource identification, competency, awareness, information sharing, documented information, and emergency preparedness and response capability.

Third step 'verification' performs evaluation - observation, measurement, analysis and evaluation; internal audit and management review.

Fourth step 'action' continuously improves the suitability, adequacy and effectiveness of the environmental management system.

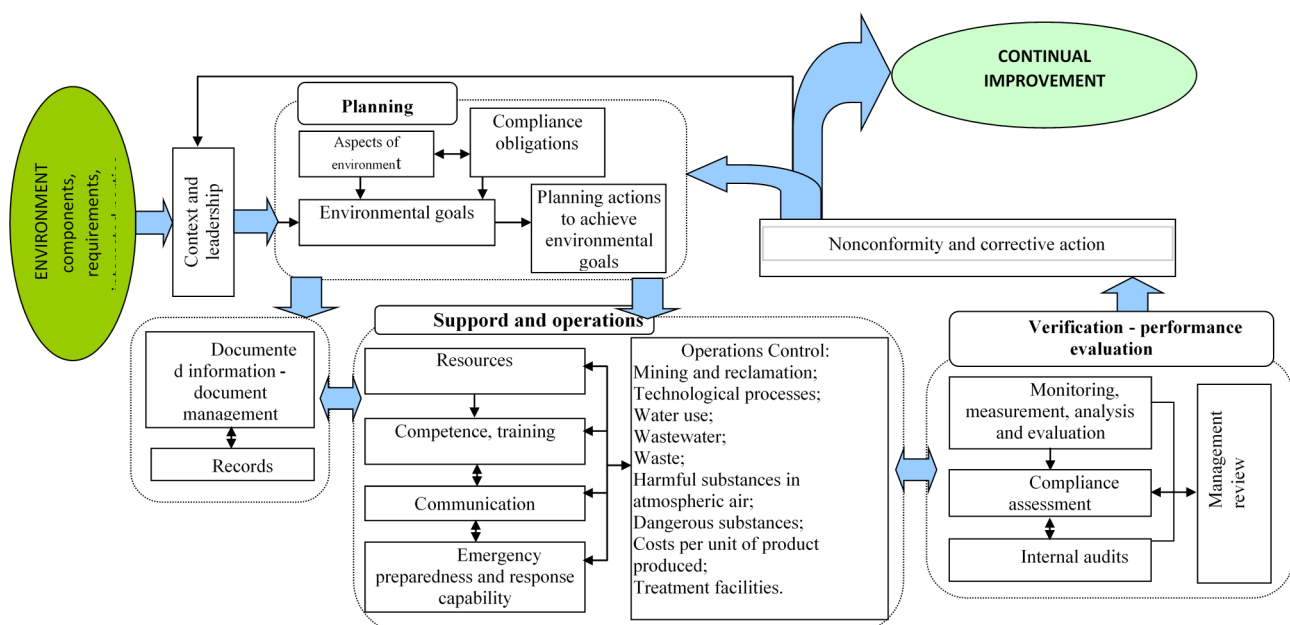


Fig. 1. Elements of the Environmental Management System.

Kaolin EAD continuously improves the suitability, adequacy and effectiveness of its environmental management system to enhance environmental performance. The organisation takes into account the results of the analysis and assessment, as well as the outputs from the management review, and determines whether there are needs or opportunities that should be addressed as part of continuous improvement.

Evidence of this continuous improvement is the methodology developed and implemented at Kaolin EAD for self-assessment of the Environmental Management System using a maturity model as a factor for achieving sustainable success.

3. Methodology for self-assessment of the environmental management system using a maturity model

Currently, there is a quality management system maturity model developed according to ISO 9004:2018 that provides guidance and addresses the ability of organizations to achieve sustainability. No maturity model has been developed for environmental management systems.

Self-assessment through an environmental management system maturity model enables organisations to review their performance against defined criteria, establish their current levels of maturity, identify their strengths and weaknesses and the associated risks and opportunities for improvement.

Babekova et. al. in [8] proposed a methodology for self-assessment by determining the maturity level of the environmental management system according to ISO 14001:2015, which was applied in Kaolin EAD, a company of Industrial Minerals sector.

In addition, Babekova N. [9] has developed and proposed an algorithm of the process of self-assessment, presented by a scheme and explained in detail the steps of the process, making its implementation easier and accessible.

The organization has developed a procedure that describes the self-assessment process.

Self-assessment is based on expert judgement. The competence and weight of the expertise of each member of the self-assessment team is determined according to certain criteria presented in a check-list.

The team of experts has carried out the assessment and determined the maturity level by assessing the fulfilment of the specified criteria. The data are presented in tabular form.

After establishing the current level of maturity, areas for improvement have been identified and 22 proposals have been prepared to complete the environmental improvement plan [10]. This report has been presented to senior management for the purpose of input to Clause 9 "Performance Evaluation" Section 9.3 "Management Review".

Following the review of the EMS, senior management has approved 16 of the suggestions for improvement, which include: activities to reduce electricity consumption; water treatment; reduce air pollution; reduce negative consequences in the event of an

emergency with a negative impact on the environment; conduct training; update documents, motivate employees to contribute to improving the performance of the environmental management system. Proposals for EMS improvement adopted by senior management become part of the planned environmental targets for 2022.

To complete the overall cycle of the self-assessment methodology, an evaluation of effectiveness was carried out - the extent to which the planned activities were implemented and the intended results achieved.

The results of the one-year cycle of activities under the EMS self-assessment methodology and the performance achieved shall be presented to auditors from the certifying organisation. Any positive assessment by them, indicated in the audit report as good practice, will contribute to the company's image through the ISO 14001 certificate issued. By holding this certificate, the organisation guarantees to all stakeholders that:

- the EMS operates in compliance with all ISO 14001 and regulatory requirements,
- Everyone working in the company is trained and competent to perform their assigned tasks flawlessly;
- the conditions in which work is carried out meet European environmental requirements;
- the selection of suppliers, subcontractors and partners is based on the required criteria
- each part of the activities is strategically specified and planned;
- every aspect aims to reduce the environmental impact;
- accurately identifies and manages risk and opportunities associated with environmental aspects.

4. Conclusion

The self-assessment, through an environmental management system maturity model, enabled Kaolin EAD to establish its current level of maturity, review and complement its strengths and weaknesses and associated risks and opportunities for improvement.

The application of the Self-Assessment Methodology of Kaolin EAD is a reviewing tool of the sustainable success level, the company has reached.

The introduction and implementation of the self-assessment methodology is a management decision aimed at protecting the environment, which improves the company's image and supports the sustainable development and better financial results.

Company's reputation is an intangible asset and a subject to annual evaluation. Such assets can yield a profit above the market return rates, as the business is worth much more than net tangible assets. The capitalised value of this excess return is economic reputation.

The methodology is used as a self-assessment tool of the company's sustainable success. The results of the Methodology increase the confidence of all stakeholders and the organisation:

- assesses annually the current level of maturity and proposes actions dedicate to

improve it with focus to sustainable development of its operations;

- it operates in an environmentally friendly manner and implements an effective environmental management system, not only meeting all the requirements of the BDS EN ISO 14001 standard and regulatory requirements, but also striving for continuous improvement of the system.

References

- [1] *Maturity model*: Available at: <http://tuj.abisnesmod/BM10.htmsevents.com/>, Accessed November 21, 2022.
- [2] ISO 45001:2018. *Occupational health and safety management systems* (2018), Available at: <https://www.iso.org/standard/63787.html>, Accessed December 01, 2022.
- [3] *Integrated management system*: Available at: <https://www.consejo.bg/интегрирана-система-за-управление>, Accessed December 11, 2022.
- [4] ISO 14001:2015. *Environmental management systems* (2015), Available at: <https://www.iso.org/standard/60857.html>, Accessed January 11, 2023.
- [5] ISO 9001:2015. *Quality management systems* (2015), Available at: <https://www.iso.org/obp/ui/#iso:std:iso:9001:ed-5:v1:en>, Accessed November 11, 2022.
- [6] Babekova N. and Vitliemov P. (2020), Literature review on integrated systems for quality management, environment and health and safety at work in the mining industry, *59th Annual scientific conference of Angel Kanchev "University of Ruse"* Vol. 59, book 5.1, Ruse, Bulgaria, pp. 26-31
- [7] Babekova, N. and Vitliemov, P. (2021), The model of maturity as a factor for sustainable development of organizations, *60th Annual scientific conference of Angel Kanchev University of Ruse*, Vol. 60, book 5.1, Ruse, Bulgaria, pp. 39-45.
- [8] Babekova, N., Gueorguiev T. and Vitliemov P. (2022). A Maturity Model as a Tool for Sustainable Development of Organizations, *2022 8th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, Ruse, Bulgaria, pp. 1-4, doi: 10.1109/EEAE53789.2022.9831414.
- [9] Babekova N. (2022). An approach to develop a methodology for self-assessment of an environmental management system through a maturity model in organizations from the mineral and raw materials industry, *61th Annual scientific conference of Angel Kanchev University of Ruse*, Vol. 61, book 5.1, Ruse, Bulgaria, pp. 61-67.
- [10] National Mining Development Strategy (mineral raw material) industry (2012), Available at: <https://mi.government.bg/files/useruploads/files/strategia.pdf>, Accessed November 22, 2022.

Trajectory Analysis of Griffon Vulture in Kvarner Bay

Stella Dumenčić ^{1*}, Lucija Žužić ¹ and Barbara Breš ¹

¹ Department of Computer Engineering,
University of Rijeka, Rijeka, Croatia

*E-mail of corresponding author: sdumencic@riteh.hr

Abstract: Advancements in technology have allowed for detailed tracking of animal movements, revealing the complexity of their behaviour, and aiding in conservation efforts. The study of Griffon Vultures in the Kvarner Bay is a prime example of the importance of trajectory analysis in understanding habitat requirements, migration patterns, and disease surveillance of endangered species. This study helps in analysing their trajectories for better understanding of their behaviour. The results suggest that Griffon Vultures soar in trajectories with low straightness and low velocity which confirms the hypothesis based on their natural movement, mostly for searching for food. Every day approximately 12 hours of data were recorded, and the average distance crossed in one day was 600 meters. The calculated fractal dimension is close to 1, which suggests directed movement towards a clear target.

Keywords: Griffon Vultures, trajectories, statistical analysis.

Received 10 April 2023

Accepted 29 April 2022

Published 20 May 2023

1. Introduction

Animal movement and the trajectories they follow have been a topic of interest for diverse scientific disciplines, such as physics, biology and ecology.

The analysis of trajectories, which are recorded tracks of individual animals' movements through space and time, has been used to answer questions about animal foraging, navigation, dispersal ecology, migration and behavioural mimicry. The obtained analysis results are then used for conservation efforts, forecasting and controlling the spread of diseases and thus preventing animal extinction from happening.

Animal trajectories are calculated as a series of specific discrete locations over time and are typically limited to two spatial dimensions. After calculations, each trajectory can be visualized and analysed as a series of steps, each one having a length and a direction. The steps represent the vector between each sampled location.

In recent years, advances in technology have enabled researchers to track and monitor the movements and trajectories of animals in exceptional detail and scale, revealing new insights into the complexity of their behaviour. The movement of animals is influenced by a variety of factors that include intrinsic factors, such as physiology, behaviour and genetics, and extrinsic factors, such as environmental conditions and resource availability. These factors interact in complex ways to shape the movement and trajectories of animals, which makes it challenging to study and understand it but on the other hand it makes it a very important task for scientists to investigate.

Our research is based on Griffon Vultures that are

soaring over Kvarner Bay in Croatia. The Griffon Vulture is an endangered species that requires attention for preventing extinction. Tracking the movement of Griffon Vultures is important for better understanding their habitat requirements and migration patterns as well as disease surveillance. In addition, trajectories analysis can help with their interaction with other species.

The rest of the paper is structured as follows: In section Methods we describe the dataset and approaches for the trajectory analysis. In section Hypothesis the hypothesis is presented. In section Results we analyse the obtained results with several evaluation metrics. Finally, we conclude the paper in section Conclusion.

2. Hypothesis

The research presents a hypothesis that there is an exact methodology for a unique formal description of animals' movement, which renders different trajectories comparable in an unbiased manner. The hypothesis claims that griffon vultures soar with low velocity (under 2 m/s) and low straightness. Since griffon vultures search for food most of the time, the hypothesis is that the fractal dimension will be ~ 1 since their movement has little deviations from their desired path.

3. Materials and methods

The R programming environment for statistical computing is a suitable option for analysing animal trajectories due to its powerful statistical packages, ability to handle large datasets, and open-source nature with online resources available. In this paper, functions from the *trajr* library developed for working with trajectories in R were used to evaluate the hypothesis that was made regarding the features of trajectories depicting the movement of Griffon Vultures.



3.1. Dataset

The dataset consists of flight records of Griffon Vultures flying over Kvarner Bay, mostly over the islands Krk and Cres, from the MoveBank website. The file is part of the research done for the LifeTrack Griffon Vulture Croatia study. The goal of the study was to follow re-released Vultures from the care center in Beli, on the island of Cres, and measure their flight behaviour [1].

Of the many values recorded in one measurement, we only focused on UTM (Universal Transverse Mercator) easting and UTM northing coordinates as they are used for creating a trajectory, as well as the timestamp. The timestamp is optional, but it is necessary if an accurate calculation of speed and acceleration for the trajectory is needed. We used UTM easting and UTM northing coordinates instead of latitude and longitude, which were also available in the recorded measurements because the functions in the trajr library only work with UTM easting and UTM northing coordinates. Conversion of latitude and longitude into UTM easting and UTM northing coordinates is also possible, but it was not necessary since the recorded data already contained both values. We used latitude and longitude to draw a line on a map representing the movement of the Griffon Vulture to better illustrate the data we are working with.

We decided to group the measurements by date and consider them a single trajectory since our goal was to observe the daily habits and routines of Griffon Vultures. Records were sorted by date in ascending order, and then by time in ascending order.

Before a trajectory can be created from UTM easting and UTM northing coordinates, the timestamp that they are linked to must be converted from a date and time format into a single number that represents the elapsed time in seconds from the chosen starting point. This is necessary to correctly estimate speed and acceleration for the trajectory. Milliseconds were not recorded.

3.2. Trajectory features

For each trajectory, diffusion distance, straightness, duration of travel, mean travel velocity, and fractal dimension is calculated. These features of each trajectory were statistically analysed, as described in the rest of this document. The preprocessing of data, the calculation of the described trajectory features, and the subsequent division of the trajectory with the largest diffusion distance into hour-long trajectories to observe the habits of Griffon Vultures in a single day is described in Figure 1.

1) Velocity

Velocity can only be measured on a segment of the trajectory between two points. There are three ways to define velocity in a single point instead of a segment. If only the velocity of the segment preceding the point is considered, we need to specify this in the function TrajVelocity from the trajr library using "backward" as the value of the diff parameter. If only the velocity of the segment following the point is considered, we use "forward" as the value of the diff parameter. If the arithmetic average of the velocity of the segment preceding the point and the velocity of the segment following the point is calculated, we use "central" as the value of the diff parameter. Since the "forward" and "backward" method can produce misleading results in case of a sudden and large change in velocity, the "central" method is the default approach if the diff parameter of the function TrajVelocity from the trajr library is not specified. We decided to use the "central" method as well due to the uncharacteristically large values for the speed of the flight of the Griffon Vulture which the "forward" and "backward" methods produced. The large values are most likely due to wind gusts, calculation errors, or measurement errors.

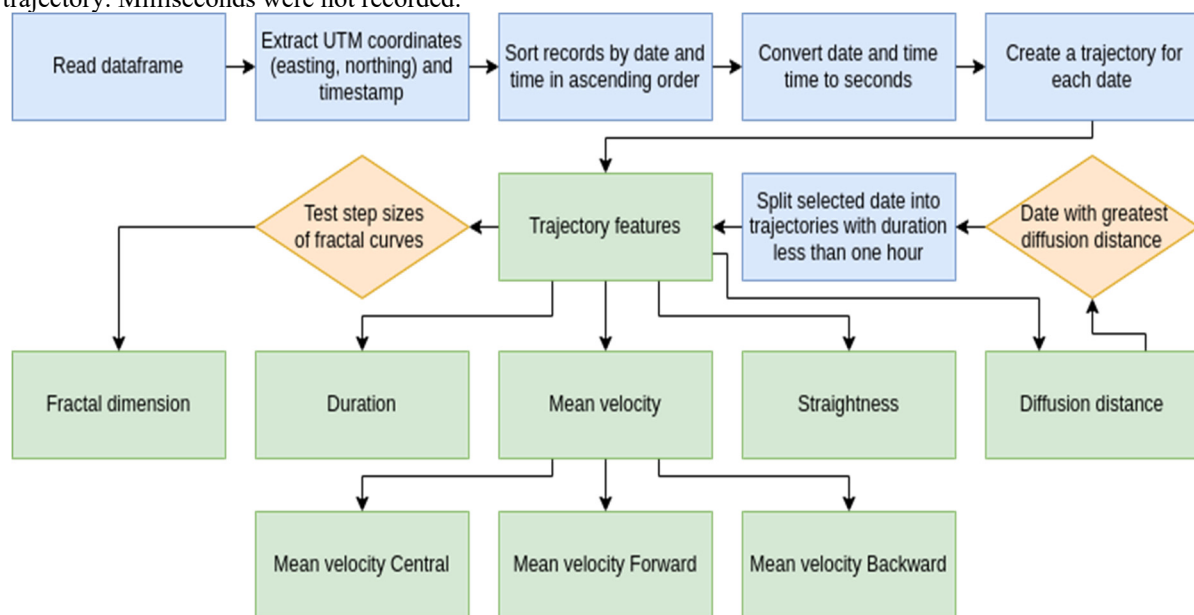


Fig. 1. Data preprocessing, calculation of trajectory features and division of the trajectory with the largest diffusion distance into hour-long trajectories.

2) Fractal dimension

The fractal dimension has been proposed as a valuable measure for straightness or tortuosity, with values ranging between 1 (representing straight line movement) and 2 (movement that frequently changes direction). However, research has shown that using fractal dimension as a measure for animal trajectories is problematic because animal trajectories are not fractal curves. Because of this their fractal dimension strongly depends on the variable range of step sizes used for the calculation [b2]. Despite these limitations, fractal dimension is still commonly calculated. The modified 'dividers' method [3] is used by default in the TrajFractalDimension function to adjust for truncation error and the corresponding equation for the 'dividers' method is contained in equation 1. By recommendation of Nams [4], this function compensates for the overestimation of D by default. This is done by walking the dividers backward and forward. After the final step, the length of the path that remains is estimated.

$$D_0 = \lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log \frac{1}{\varepsilon}} \quad (1)$$

In order to calculate fractal dimension, a range of step sizes that will be used for all trajectories must be defined. The TrajLogSequence function from the library trajr was used to return a sequence of 10 points which are regularly spaced between 0.1 and 3 when plotted on a logarithmic axis. Only 10 points were used because computational complexity and execution time increases significantly with a larger number of points. A test must be conducted to ensure that the trajectory is a fractal curve for the given step sizes. This is done by using the function TrajFractalDimensionValues which returns path lengths for a given trajectory and a range of step sizes. If the relationship between step size and path length is linear, then the trajectory is a fractal curve for the given range of step size. With a process of trial and error, we settled on the range of step sizes between 0.1 and 3 because the relationship between step size and path length was linear for all tested trajectories in this range.

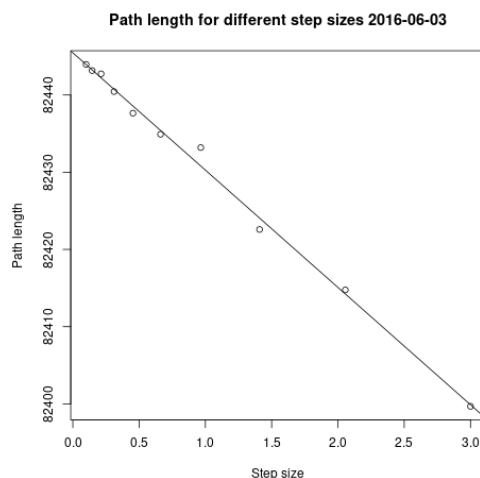


Fig. 2. Relationship between step size and path length.

In order to demonstrate this, one curve representing the relationship between step size and path length was plotted, as shown in Figure 2, along with a 45-degree reference line, for a trajectory that covers a whole day and has the largest diffusion distance. This can be seen in Figure 2. For larger step sizes, nonlinearity was observed. Additionally, some short trajectories could not support a step size larger or equal to 4.

3) Straightness

Trajr contains several built-in methods for measuring the straightness or tortuosity of trajectories. One of these methods uses the simple D/L formula, where D is the distance between the trajectory's start and end, while L is the trajectory's length. The TrajStraightness function calculates the straightness index in this manner. The straightness index obtained with this function ranges from 0 to 1, with 1 indicating a completely straight line. Some sources [5] consider this value to be a reliable indicator of efficiency if applied to a directed walk, even though it is not applicable to random trajectories. Others [6] approximate the straightness index with the length of the mean vector of turning angles (r) after rediscritization to a constant step length.

As animal movement trajectories are not random trajectories, but are usually directed walks towards some goal in the environment, the simpler straightness index defined by D/L and calculated by the function TrajStraightness, was used in this paper.

3.3. Statistical analysis

The minimum, first quartile, median, third quartile, maximum, standard deviation, and variance are calculated to help estimate a continuous distribution from the data in further processing. The statistical analysis of each trajectory feature with all the calculated parameters and estimated distributions is described in Figure 3.

3.4. Estimating a continuous distribution

Histograms are created for each feature of the trajectories. A line representing the estimated kernel density is drawn along with the histogram. Lines that represent the probability density of the estimated normal, log-normal, exponential, Gamma, and Weibull distributions are also drawn. Before a continuous distribution can be estimated, duplicate values must be removed from the data.

Any missing and infinite values must be removed as well. We test if the data fits the estimated continuous distributions using the Shapiro-Wilk test for the normal distribution and the Kolmogorov-Smirnov test for the normal as well as all the other estimated distributions. If the p-value is less than the alpha value of 0.05, the initial hypothesis that the data fits the estimated continuous distribution is rejected.

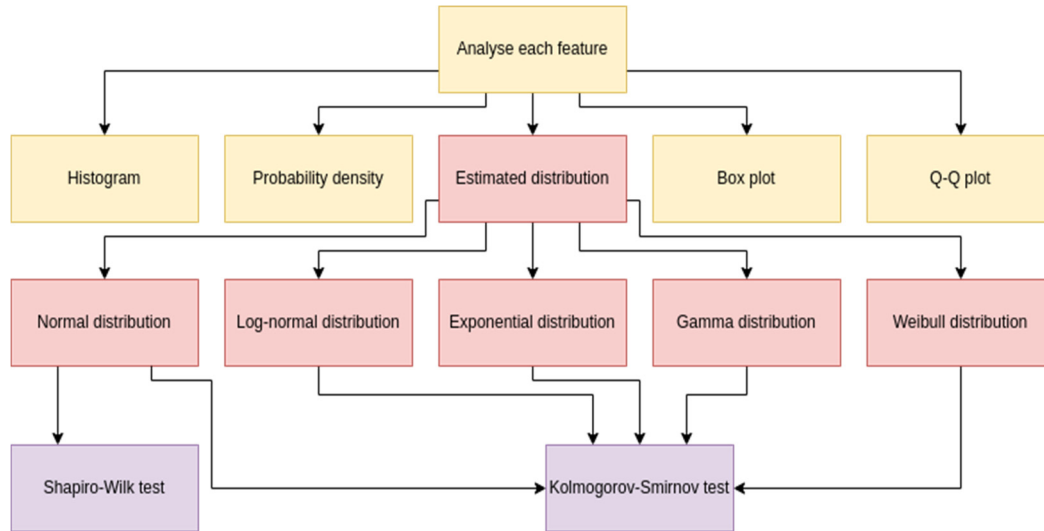


Fig. 3. Statistical analysis of trajectory features with calculated parameters and estimated distributions.

1) The log-normal distribution

In the log-normal distribution, the logarithm of the variable is normally distributed. Thus, $\exp(\mu)$ equals the median of the data, and $\exp(\mu + \frac{\sigma^2}{2})$ equals the mean of the data. We can calculate the median and the mean of the data and use these values to calculate μ and σ for the estimated log-normal distribution, as illustrated in the relations established in the previous sentence.

2) The exponential distribution

In the exponential distribution, the inverse value of the rate parameter is equal to the mean of the data. The inverse value of the rate parameter is equal to the scale parameter. We can use this knowledge to estimate the rate (λ) and the scale parameter for the estimated exponential distribution.

3) The Gamma distribution

In the Gamma distribution, the mean value of the data is equal to the product of the shape (k) and scale (θ) parameters, or $k\theta$. The variance of the Gamma distribution is equal to the product of the shape parameter and the square of the scale parameter, or $k\theta^2$. We can calculate the variance and the mean of the data and use these values to calculate the shape and scale parameters for the estimated Gamma distribution, as explained in the relations established in the previous sentence.

4) The Weibull distribution

Instead of a step-by-step calculation, the function `eweibull` from the `EnvStats` library was used to estimate the shape (k) and scale (λ) parameters for the Weibull distribution in order to simplify the code.

3.5. Analysis of hour-long trajectories

After the analysis was complete for all the trajectories that cover a full day, the trajectory which has the largest diffusion distance was identified. This trajectory, which covers a full day, was divided into smaller trajectories. The motivation for creating shorter trajectories was to determine whether there was any change in the activity of Griffon Vultures within a single day, and not exclusively

over the course of several days. We chose the date with the largest diffusion distance because this would give us more data to analyse. This analysis was not repeated for each day because of time constraints. The goal was to choose smaller trajectories that have a duration of approximately one hour. Any larger duration would be too close to a full day. Any shorter duration would have a very small number of points and a very small diffusion distance, so calculated velocity and other values would not be accurate and would not reflect the data as a whole.

These small trajectories were created by adding points to a trajectory one by one until the duration of the trajectory became larger or equal to 3500 seconds, at which point the created trajectory was saved and a new one began. This was done in order to create trajectories that had a duration as close as possible to one hour. If we had chosen to terminate trajectories once the duration exceeded or became equal to exactly 3600 seconds, the duration of each trajectory would have been significantly larger than the desired hour. This happens because the average time between measurements is quite large when compared to an hour, usually several minutes. Once these new shorter trajectories are created, the same features are calculated and analysed in the same way it was done for the longer trajectories.

4. Results

The results obtained from the analysis of trajectories confirmed our thesis on the movement of griffon vultures. Griffon vultures move by soaring, and this is evident from the results that show a low speed of movement. The outliers in speed show that anomalies, such as strong wind, can increase the soaring velocity. It can also be seen that the movement trajectories of griffon vultures have low straightness which confirmed the thesis of circular soaring for finding food. The fractal dimension is ~ 1 , which means that the griffon vultures movement did not deviate from their desired path very often. The duration of daily trips is measured for ~ 12 hours. The graphical representation of trajectories is shown in Figure 4 where the blue line shows the trajectory with the largest diffusion distance and the red line showing all trajectory points of all days.

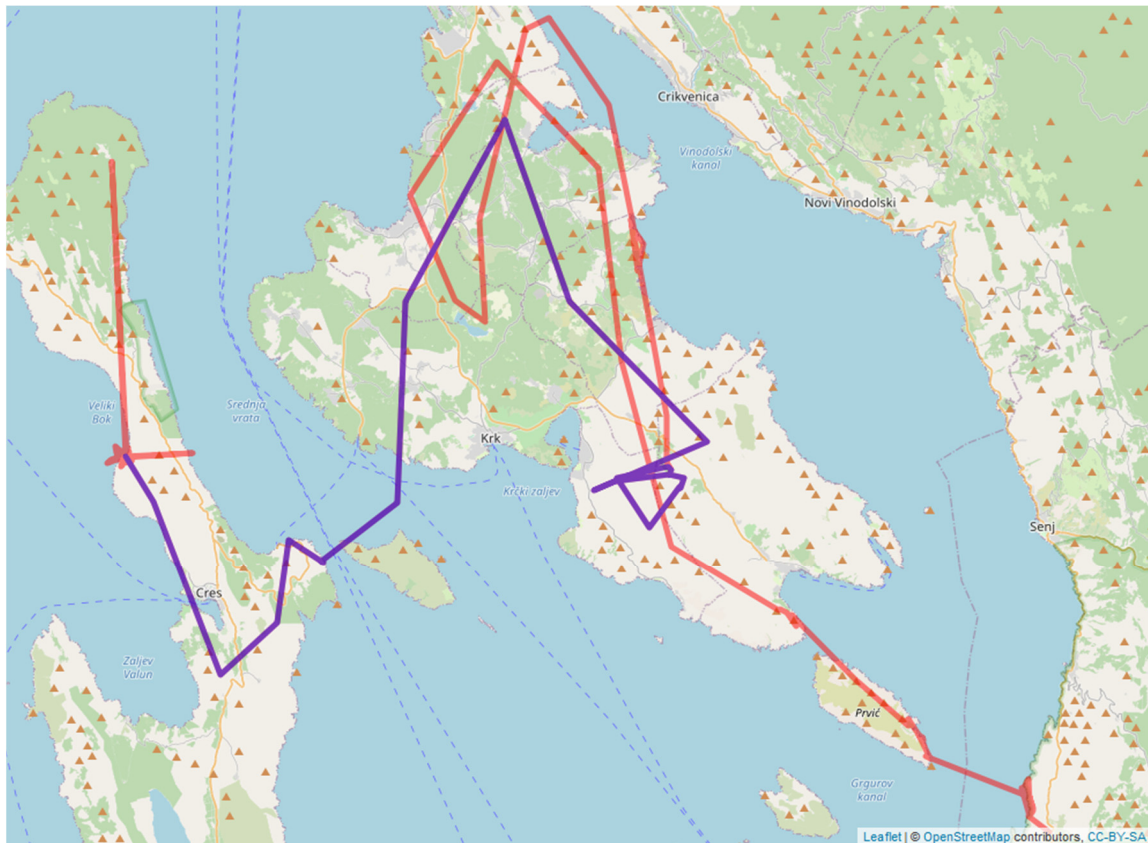


Fig. 3. Griffon Vultures trajectories. Blue showing the trajectory with the largest diffusion distance and red showing all trajectory points of all days.

The statistical analysis of all trajectories can be found in Table 1 and the statistical analysis of the specific date 2016-06-03 can be found in Table 3.

In Table I it can be seen that Griffon Vultures fly almost 600 meters on average with a trajectory that has

an average straightness of 0.8125. The central velocity shows that the average speed of Griffon Vultures in the Kvarner Bay area is less than 1 m/s which confirms the hypothesis of them flying at low speed.

Table 1. Observed features of trajectories and p-values.

Name	Min	Q1	Median	Arithmetic mean		Q3	Max	Variance	St. dev.
Diffusion distance (m)	0.78358	4.35953	7.17817	587.2926	11.80907	25526.165	9896147	3141.52	
Straightness	0.0001	0.00381	0.00965	0.08125	0.03241	0.98585	0.03444	0.18558	
Duration	28782	41375.5	41375.5	41124.73	41391.25	53973	10895691	3300.862	
Mean velocity (central) (m/s)	0.00013	0.01523	0.04085	0.53539	0.14121	52.96057	22.38886	4.73167	
Mean velocity (forward/backward) (m/s)	0.00022	0.02106	0.07847	0.13891	0.16444	1.97052	0.06416	0.25330	
Fractal dimension	1.00011	1.03526	1.04811	1.04811	1.06461	1.11115	0.00053	0.02312	

Table 2. Observed features of trajectories and p-values.

Distribution	Normal	Normal	Log-normal	Exponential	Gamma	Weibull
Test	Shapiro-Wilk		Kolmogorov - Smirnov			
Diffusion distance (m)	< 2.2e-16	< 2.2e-16	4.848e-8	< 2.2e-16	< 2.2e-16	4.808e-11
Straightness	< 2.2e-16	< 2.2e-16	0.00197	< 2.2e-16	2.472e-11	0.00816
Duration	2.425e-10	1.579e-7	8.677e-9	< 2.2e-16	3.888e-7	1.09e-7
Mean velocity (central)	< 2.2e-16	< 2.2e-16	0.02924	< 2.2e-16	< 2.2e-16	0.00947
Mean velocity (forward/backward)	< 2.2e-16	1.103e-9	0.00301	0.00557	8.161e-10	0.6163
Fractal dimension	0.0214	0.5864	0.00558	< 2.2e-16	0.5525	0.6307

Table 3. Statistical analysis of trajectory features for 2016-06-03.

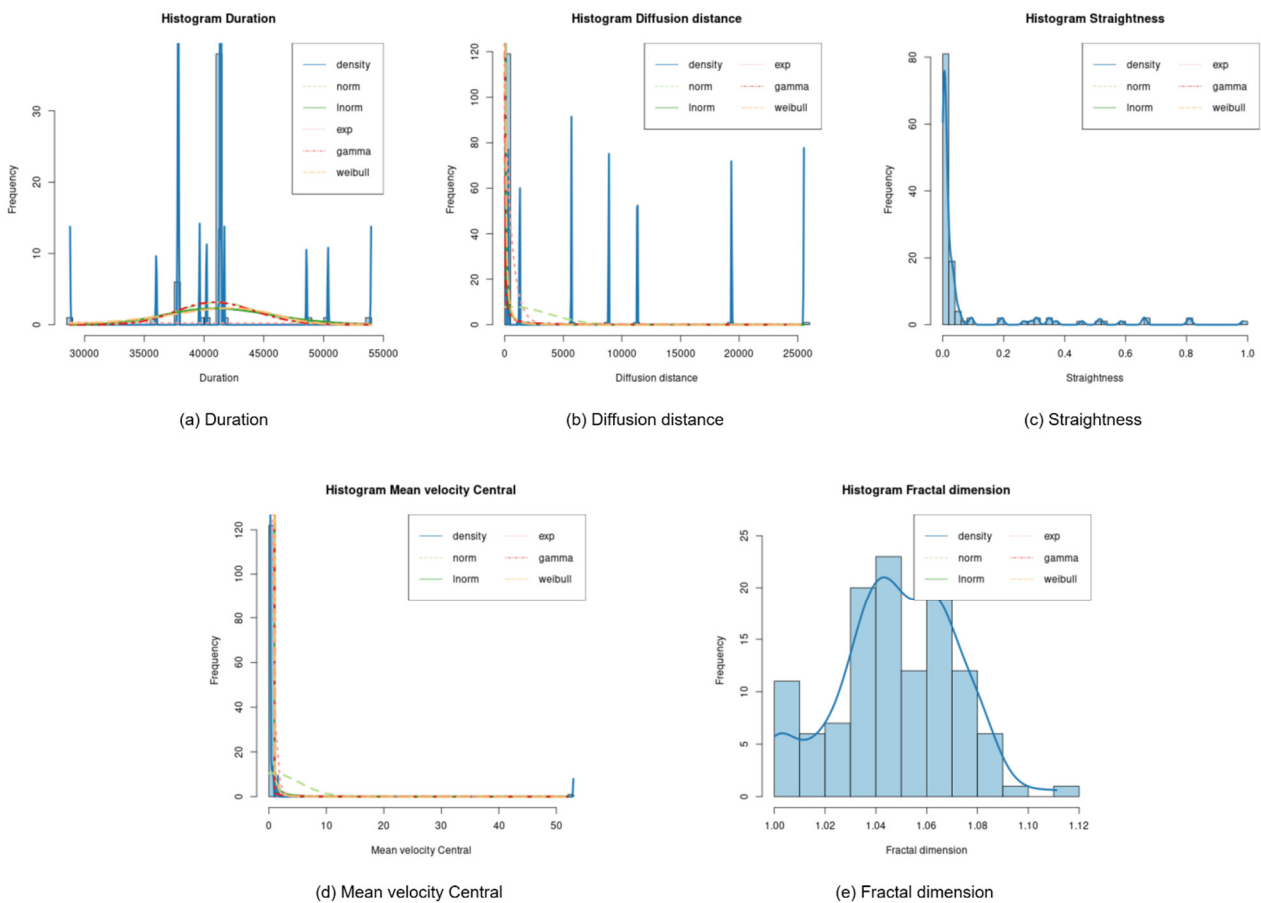
Name	Min	Q1	Median	Arithmetic mean	Q3	Max	Variance	St. dev.
Diffusion distance (m)	7.2619	35.61172	2434.5412	3021.966	2824.16292	12404.78795	16060629	4007.571
Straightness	0.24061	0.42038	0.68374	0.66793	0.98122	1.0000	0.09571	0.30937
Duration	1770.00	3590.50	3600.50	3417.2	3607.75	3622.00	335189.1	578.9552
Mean velocity (central) (m/s)	0.00487	0.01223	0.59749	1.21629	1.56179	5.521138	3.04297	1.74441
Mean velocity (forward/backward) (m/s)	0.00486	0.02431	0.72097	1.55584	2.1577	7.96301	6.08095	2.46595
Fractal dimension	1.00000	1.00007	1.00022	1.00781	1.00404	1.0443	0.00024	0.01542

Table 4. Observed features of trajectories and p-values for 2016-06-03.

Distribution	Normal	Normal	Log-normal	Exponential	Gamma	Weibull
Test	Shapiro-Wilk	Kolmogorov - Smirnov				
Diffusion distance (m)	0.004759	0.2324	0.05899	0.08191	0.2252	0.3083
Straightness	0.07501	0.6423	0.08154	0.2317	0.5789	0.5741
Duration	1.834e-7	0.00741	0.00919	0.00241	0.00459	0.00457
Mean velocity (central)	0.00324	0.5163	0.06521	0.0885	0.3868	0.6829
Mean velocity (forward/backward)	0.00047	0.212	0.08607	0.1104	0.8097	0.7359
Fractal dimension	4.57e-5	0.07513	0.00791	0.00023	0.07691	0.09094

For the fractal dimension, since the griffon vultures have an intended destination in mind and do not alter the

direction of their movement, even the maximal value is near 1.

**Fig. 5.** Trajectory features histograms.

In the Table 3, the date with the longest diffusion distance is presented which is ~12405 meters. The straightness is, as in the last table, a value that corresponds with low straightness, and in this case with a maximal value of 1. It can be seen, that on this day, the average Central velocity is low, with an average value of 1.2 m/s. The fractal dimension has a maximal value of 1.0433 and thus confirms that the movement does not change course significantly.

Based on the fact that a lot of natural phenomena behave according to a distribution, we performed 6 tests for 5 different distributions to check if the griffon vulture movement will be distributed to one of these distributions: Normal, Log-normal, Exponential, Gamma and Weibull. For the normal distribution we performed the Shapiro-Wilk and Kolmogorov-Smirnov test, while for all other distributions, the performed test is the Kolmogorov-Smirnov test.

The results of the observed characteristics of the trajectories and their p-values are presented in Table 2, while the observed characteristics and their p-values of the date 2016-06-03 are shown in Table 4.

The bold values of the results in the given table indicate that the hypothesis H_0 is not rejected, i.e. that the p-value is greater than 0.05.

The histograms for each trajectory feature can be found in Figure 5.

5. Discussion and conclusion

Based on our research, the Griffon Vulture movement in the Kvarner Bay wasn't yet statistically analysed, so the importance of this paper is to show their movement trajectories which could help in studying them and monitoring changes with the goal of saving the species.

The trajectory analysis of Griffon Vultures shows that they usually move slowly, searching for food on the ground with circular trajectories. However, their movement speed can increase with strong wind.

In the future, this work can be extended to compare the flight behavior of Griffon Vultures in the Kvarner Bay with the flight behavior of Griffon Vultures in the Alps and Himalayan Vultures. The project could be expanded to include a graphical interface where users can select a date from a drop-down menu and download the data described in the paper for that chosen date. The downloadable data would include the values obtained by statistical analysis in a tabular format as well as all the generated charts in the form of images.

References

- [1] <https://www.movebank.org/>, Accessed April 30, 2023.
- [2] Cheung A., Zhang S., Stricker, C. and Srinivasan M. V. (2007). Animal navigation: the difficulty of moving in a straight line. *Biological Cybernetics*, 97, pp. 47–61. <https://doi.org/10.1007/s00422-007-0158-0>
- [3] Sugihara G. and May R. M. (1990). Applications of fractals in ecology, *Trends in Ecology & Evolution*, vol. 5 issue 3, pp, 79-86. doi:10.1016/0169-5347(90)90235-6.
- [4] Nams V. O. (2006). Improving Accuracy and Precision in Estimating Fractal Dimension of Animal movement paths. *Acta Biotheoretica*, vol. 54, pp. 1-11. <https://doi.org/10.1007/s10441-006-5954-8>.
- [5] Benhamou S. (2004). How to reliably estimate the tortuosity of an animal's path. *Journal of Theoretical Biology*, vol. 229, issue 2, pp. 209-220, <https://doi.org/10.1016/j.jtbi.2004.03.016>
- [6] Batschelet, E. (1981). *Circular statistics in biology*. Academic Press, New York, NY, USA, p. 371

